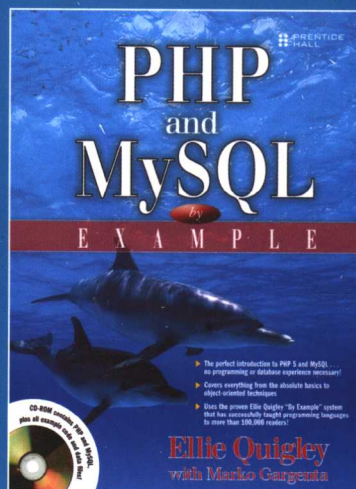


From Programmer, For Programmer

- 采用成功的“案例剖析”
(By Example) 教学方式
- 内容全面, 从基本概念到
面向对象技术
- 无需任何经验, 快速掌握
数据库驱动的动态Web
开发



PHP与 MySQL 案例剖析

[美] Ellie Quigley Marko Gargenta 著
王军 等 译

TP311.138/481

2007

From Programmer,
For Programmer

PHP与 MySQL案例剖析

[美] Ellie Quigley Marko Gargenta 著
王军 等 译

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

PHP 与 MySQL 案例剖析 / (美) 奎格利 (Quigley, E),
(美) 加根塔 (Gargenta, M.) 著; 王军等译. —北京: 人民
邮电出版社, 2007.12

ISBN 978-7-115-16825-2

I. E… II. ①奎…②加…③王… 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 (2007) 第 142556 号

版权声明

Authorized translation from the English language edition, entitled PHP AND MYSQL BY EXAMPLE, 1st Edition, 0131875086 by QUIGLEY, ELLIE; GARGENTA, MARKO, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2007 Pearson Education, Inc.

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 Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2007.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签。无标签者不得销售。

PHP 与 MySQL 案例剖析

- ◆ 著 [美] Ellie Quigley Marko Gargenta
译 王 军 等
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>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16
印张: 42.75
字数: 1 047 千字 2007 年 12 月第 1 版
印数: 1—5 000 册 2007 年 12 月北京第 1 次印刷

ISBN 978-7-115-16825-2/TP

定价: 85.00 元

读者服务热线: (010)67132075 印装质量热线: (010)67129223

Arspract

内 容 提 要

本书通过大量的实例详细说明了如何使用 PHP 和 MySQL 轻松地开发实用的动态 Web 站点，内容详实、全面，是一本非常出色的入门书及参考手册。本书详细讨论了 PHP 的基本语法、数据类型、常用内置函数，还介绍了如何创建自定义函数、访问文件和目录、处理表单数据、使用 Cookie 和会话，着重说明了 PHP 如何与 MySQL 配合工作。

本书可以帮助读者从最简单的 PHP 脚本和 SQL 查询技术开始学起，逐渐掌握如何使用 PHP 和 MySQL 构造数据库驱动的动态 Web 站点。本书不仅适合 Web 开发人员，也适合编程经验很少的 Web 设计人员。

Acknowledgments

致 谢

很多人对本书有所贡献。我要感谢长期合作的 Prentice Hall 的编辑 Mark L. Taub, 世界上最有天赋的排版员 Vanessa Moore, 伟大的主编 Julie Nahil。另外, Matthew Leingang、Sander van Zoest、David Mercer 和 Jason Wertz 为检查原稿提供了很大帮助; 如果本书还有什么不当之处, 都是我的责任。

我还要感谢提供了实验数据的学生, 包括 Rita McCue、Sanjay Shahri、Ryan Belcher、Debra Anderson 和 Catherine Nguyen。

书中漂亮的插图是由 Elizabeth Staechelin 和 Daniel Staechelin 创建的。另外还要感谢为画廊范例提供作品的画家们, 他们是 Elliott Easterling、Laura Blair、Stuart Sheldon 和 Todd Brown。

Ellie Quigley
于加利福尼亚旧金山
2006 年 9 月

Preface

前 言

在过去的几年里，学习 Perl/CGI 课程的学生不断询问我何时从 CGI 转移到 PHP，以及我是否应该教授 PHP 课程或是编写有关 PHP 范例的图书。对此我起初并没有太在意，直到几年前在旧金山参加了一个 PHP/MySQL 培训班。我的授课老师是 Marko Gargenta，他让我有了编写本书的灵感。在一次午餐时，我对他说上课时坐在我旁边的女孩是个编程经验很少的 Web 设计员，她担心不能跟上课程，向我询问哪里可以找到为设计人员而不是程序员编写的 PHP 图书。Marko 也从学生中听到了类似的想法，于是我们探讨了如何解决这个问题，从而埋下了写作这本书的种子。

虽然从理论上来说，Web 设计人员或开发人员应该不需要 PHP 编程经验就可以改变页面的内容，而程序员应该只关心逻辑问题（比如计算、向数据库发送数据等），但是一般来说，他们并不是各干各的。举例来说，有这样一个页面，当用户在一个 HTML 表单里输入银行信息之后，一个 PHP 程序就进行一些计算，在发现资金不足时就以红色粗体字发送错误消息。在这种情况下，PHP 和 HTML 是集成在一起的，前者进行计算和生成错误消息，后者以粗体字显示消息。也许让设计与程序逻辑区分开是我们的目标，但面对当今复杂的 Web 开发，这通常是不可能实现的。

另外还有一个数据库管理系统的问题。在哪里保存处理后的数据？谁设计数据库及其表格？谁管理它？从页面获得的数据如何传送到 PHP 程序，然后再传送到数据库？答案之一是使用 MySQL。难道这又是另外一个世界吗？

我第一次与 Marko 探讨上述问题之后，就决心把这些技术集成在一起。当 Prentice Hall 同意出版我们的图书时，整本书的内容安排是有相当难度的。在初稿完成之后，我根据初稿的 PDF 版本开设了“PHP 和 MySQL 编程入门”的课程。我注意到来听课的 Web 设计人员比程序员多，而且他们一般都担心难以接受这些内容。但是通过真实的范例以及我们提供的实验室，他们从一开始就感受到了成功的喜悦。我们欣喜地看到设计人员和程序员一起分享他们的经验，抛弃了分隔其工作环境的人为障碍。

本书的目标在于初步但完整地介绍如何联合使用 PHP 与 MySQL，从而让静态 HTML 变成动态的。本书介绍的实验环境与练习都经过我本人、Marko 或我们学生的测试。我相信无论读者在完成 Web 站点的哪种支持工作，或是刚刚开始设计自己的站点，都会发现本书是一本实用且完整的指导书。

书中部分 PHP 和 MySQL 源代码可以在人民邮电出版社的网站上找到（先在首页上查找本书）。网址为 www.ptpress.com.cn。

Contents

目 录

第 1 章 简介	1	2.4.1 在命令行运行 PHP	24
1.1 从静态到动态站点	1	2.4.2 确定 PHP 的版本	25
1.1.1 静态 Web 站点	1	2.4.3 在命令行执行脚本	25
1.1.2 动态 Web 站点	2	2.4.4 以交互方式运行 PHP	26
1.1.3 什么是开放源码	3	2.4.5 命令行 PHP 的选项	26
1.2 关于 PHP	3	2.4.6 Php.ini 文件	27
1.3 关于 MySQL	6	2.5 本章小结	29
1.3.1 在哪里获得 MySQL 和 文档	6	2.5.1 应该掌握的内容	29
1.3.2 MySQL 特性	7	2.5.2 下一步	29
1.3.3 如何安装 MySQL 和 PHP	7	第 3 章 PHP 快速入门	31
1.3.4 MySQL 和 PHP 的优点	8	3.1 快速入门、快速参考	31
1.4 本章小结	9	3.1.1 对于程序员	31
1.4.1 应该掌握的内容	9	3.1.2 对于非程序员	31
1.4.2 下一步	9	3.1.3 PHP 语法和结构	31
第 2 章 入门	10	3.2 本章小结	44
2.1 Web 页面的生存周期	10	第 4 章 构件：数据类型、文字、 变量和常数	45
2.2 PHP 脚本剖析	11	4.1 数据类型	45
2.3 需要考虑的一些事情	16	4.1.1 数值	46
2.3.1 PHP 和 HTML 是不同的 语言	16	4.1.2 字符串	47
2.3.2 语句、空白和换行	17	4.1.3 布尔值	52
2.3.3 注释	19	4.1.4 特殊数据类型	52
2.3.4 使用 PHP 函数	21	4.2 变量	54
2.4 复习	24	4.2.1 定义和赋值	54
		4.2.2 变量名	54

4.2.3	变量声明和初始化	55	5.1.15	执行操作符	110
4.2.4	显示变量	57	5.1.16	错误控制操作符	111
4.2.5	变量和混合数据类型	60	5.1.17	类型操作符	112
4.2.6	串接和变量	61	5.2	本章小结	112
4.2.7	引用	62	5.2.1	应该掌握的内容	112
4.2.8	可变变量(动态变量)	63	5.2.2	下一步	113
4.2.9	变量的作用域	64	第 6 章	字符串	114
4.2.10	管理变量	65	6.1	字符串是什么	114
4.2.11	表单变量简介	69	6.1.1	引号	115
4.3	常数	76	6.1.2	字符串操作符	117
4.3.1	常数是什么	76	6.2	字符串函数	119
4.3.2	利用 define() 函数创建 常数	76	6.2.1	字符串的格式化和显示	119
4.3.3	constant() 函数	77	6.2.2	数字和货币的格式化	123
4.3.4	预定义常数和一些“神奇” 常数	78	6.2.3	字符串长度	124
4.4	本章小结	79	6.2.4	字符串里的单词数量	125
4.4.1	应该掌握的内容	79	6.2.5	改变字符串大小写	126
4.4.2	下一步	80	6.2.6	字符串比较	128
第 5 章	操作符	82	6.2.7	字符串相似性	133
5.1	关于 PHP 操作符和表达式	82	6.2.8	分解字符串	137
5.1.1	赋值	83	6.2.9	重复字符串	138
5.1.2	优先与结合	83	6.2.10	字符串剪裁与填充	139
5.1.3	算术操作符	86	6.2.11	查找和替换	142
5.1.4	短路赋值操作符	87	6.2.12	字符定位	145
5.1.5	自动增量和自动减量 操作符	88	6.2.13	提取字符串片断—— 子串	146
5.1.6	一些常用数学函数	90	6.2.14	特殊字符和字符串	153
5.1.7	强制转化操作符	91	6.2.15	处理 HTML 特殊字符	161
5.1.8	串接操作符	93	6.3	其他字符串函数	164
5.1.9	比较操作符	94	6.4	本章小结	167
5.1.10	数字比较	96	6.4.1	应该掌握的内容	167
5.1.11	比较字符串	97	6.4.2	下一步	167
5.1.12	逻辑操作符	98	第 7 章	条件与循环	169
5.1.13	条件操作符	106	7.1	控制结构、语句块和复合语句	169
5.1.14	比特操作符	108	7.2	循环	177
			7.2.1	while 循环	177

7.2.2	do/while 循环	179	9.1	什么是函数	262
7.2.3	for 循环	180	9.1.1	函数声明、定义和调用	263
7.2.4	foreach 循环	183	9.1.2	传递变元	265
7.2.5	用 break 和 continue 控制 循环	183	9.1.3	返回值	273
7.3	本章小结	187	9.1.4	使用回调函数	279
7.3.1	应该掌握的内容	187	9.1.5	作用域	282
7.3.2	下一步	187	9.1.6	嵌套函数	286
第 8 章	数组	190	9.1.7	递归函数	288
8.1	什么是数组	190	9.1.8	函数库——要求与包含	290
8.1.1	创建和命名数组	193	9.2	本章小结	293
8.1.2	访问数组的元素(值)	197	9.2.1	应该掌握的内容	293
8.1.3	显示数组内容	201	9.2.2	下一步	294
8.1.4	利用循环访问数组元素	204	第 10 章	PHP 表单	295
8.1.5	检查数组是否存在	210	10.1	简介	295
8.1.6	从数组创建字符串和 从字符串创建数组	212	10.2	复习 HTML 表单	295
8.1.7	数组的大小	214	10.2.1	浏览器担当的角色	295
8.1.8	从数组提取关键字和值	216	10.2.2	服务器担当的角色	298
8.1.9	从数组元素创建变量	221	10.2.3	创建 HTML 表单	299
8.1.10	多维数组	226	10.3	PHP 与表单	303
8.1.11	数组排序	230	10.3.1	register_globals 指令	304
8.1.12	随机处理数组	237	10.3.2	使用超全局数组获取表 单数据	305
8.2	修改数组(清除、删除、添加 和改变元素)	240	10.3.3	请求方法	306
8.2.1	删除数组及其元素	241	10.3.4	表单字段名称及用户 输入里的怪字符	306
8.2.2	向数组添加元素	244	10.3.5	表单参数: \$_REQUEST 数组	308
8.2.3	拷贝数组的元素	248	10.3.6	表单参数和中样式	311
8.2.4	结合与合并数组	250	10.3.7	表单参数和长(旧) 样式	316
8.2.5	数组操作符	254	10.3.8	处理具有多选项的表单	317
8.2.6	其他数组函数	257	10.3.9	使用图像按钮的表单	320
8.3	本章小结	260	10.3.10	自处理 HTML 表单	322
8.3.1	应该掌握的内容	260	10.3.11	使用隐藏字段	324
8.3.2	下一步	261	10.3.12	用户的重定向	326
第 9 章	自定义函数	262	10.3.13	上传文件	329

10.3.14 粘性表单	333	12.1 什么是正则表达式	387
10.3.15 何处获取关于超全局 数组的信息	337	12.2 式样匹配函数	388
10.3.16 如何获取服务器信息	337	12.2.1 查找式样	389
10.3.17 如何获取环境的信息	340	12.2.2 查找和替换	394
10.4 本章小结	341	12.2.3 获得控制——正则 表达式元字符	403
10.4.1 应该掌握的内容	341	12.2.4 在文本文件里查找 式样	431
10.4.2 下一步	342	12.2.5 利用 PHP 实现表单 检验	433
第 11 章 文件和目录	344	12.2.6 获取 Web 上的帮助	438
11.1 文件	344	12.3 本章小结	439
11.1.1 文件权限和所有权	344	12.3.1 应该掌握的内容	439
11.1.2 UNIX/Linux 权限	344	12.3.2 下一步	440
11.1.3 Windows 权限	346	第 13 章 MySQL 简介	442
11.2 Web 服务程序、PHP 和权限	348	13.1 关于数据库	442
11.2.1 PHP 内置函数	348	13.1.1 客户端/服务器数据库	443
11.2.2 文件句柄	349	13.1.2 与数据库通信	443
11.2.3 打开文件	350	13.1.3 MySQL 的优缺点	443
11.2.4 打开文件进行读取	351	13.2 剖析关系型数据库	444
11.2.5 文件指针的定位	358	13.2.1 数据库服务程序	445
11.2.6 打开 URL 进行读取	361	13.2.2 数据库	445
11.2.7 不使用文件名称读取 文件内容	362	13.2.3 表	446
11.2.8 打开文件进行写入和 添加	367	13.2.4 记录和字段	446
11.2.9 文件检查	371	13.2.5 主键和索引	448
11.2.10 创建、拷贝、重命名和 删除文件	375	13.2.6 数据库规划	448
11.3 目录	376	13.3 连接到数据库	449
11.3.1 打开和读取目录	377	13.3.1 MySQL 命令行选项	450
11.3.2 获取路径信息	378	13.3.2 图形用户工具	452
11.3.3 改变和获取当前目录	379	13.4 MySQL 授权系统	455
11.4 利用包含文件管理内容	380	13.4.1 登录到数据库服务 程序	455
11.5 本章小结	385	13.4.2 查找数据库	456
11.5.1 应该掌握的内容	386	13.4.3 “user” 表	457
11.5.2 下一步	386	13.4.4 “db” 表	458
第 12 章 正则表达式和式样匹配	387	13.4.5 “host” 表	459

13.4.6 实际范例	460	第 15 章 PHP 与 MySQL 的集成	507
13.4.7 Grant 与 Revoke 命令	461	15.1 导言	507
13.4.8 创建和清除数据库	462	15.1.1 连接到数据库服务程序	507
13.4.9 一些有用的 MySQL 函数	463	15.1.2 选择数据库	509
13.5 本章小结	464	15.1.3 执行 SQL 语句 (INSERT、 UPDATE、DELETE)	510
13.5.1 应该掌握的内容	464	15.1.4 获取查询结果 (SELECT)	512
13.5.2 下一步	465	15.1.5 其他常用 MySQL 函数	515
第 14 章 SQL 语言指南	466	15.2 留言簿范例	520
14.1 SQL 是什么	466	15.2.1 第 1 步: 设计数据库	520
14.1.1 SQL 的标准化	466	15.2.2 第 2 步: 创建消息 页面	521
14.1.2 执行 SQL 语句	467	15.2.3 第 3 步: 列出全部 消息	524
14.1.3 关于 SQL 命令/查询	468	15.2.4 范例小结	525
14.1.4 SQL 与数据库	470	15.3 本章小结	526
14.1.5 SQL 数据库表	471	15.3.1 应该掌握的内容	526
14.2 SQL 数据操作语言 (DML)	473	15.3.2 下一步	526
14.2.1 SELECT 命令	474	第 16 章 Cookie 和会话	528
14.2.2 INSERT 命令	484	16.1 什么是无状态	528
14.2.3 UPDATE 命令	484	16.2 什么是 Cookie	528
14.2.4 DELETE 语句	485	16.2.1 Cookie 的成分	529
14.3 SQL 数据定义语言	486	16.2.2 Cookie 的属性	530
14.3.1 创建数据库	486	16.3 PHP 与 Cookie	531
14.3.2 SQL 数据类型	487	16.3.1 使用 setcookie() 函数 创建 Cookie	531
14.3.3 创建表	488	16.3.2 利用 Cookie 追踪用户	536
14.3.4 创建关键字	490	16.3.3 延长 Cookie 的生命	539
14.3.5 关系	492	16.3.4 缓存与 HTTP 头标	540
14.3.6 改变表	495	16.3.5 删除 Cookie	543
14.3.7 删除表	496	16.4 什么是会话	544
14.3.8 删除数据库	496	16.4.1 在哪里保存会话	546
14.4 SQL 函数	496	16.4.2 启动基于 Cookie 的 会话	548
14.4.1 数值函数	497		
14.4.2 字符串函数	499		
14.4.3 日期和时间函数	500		
14.5 本章小结	503		
14.5.1 应该掌握的内容	503		
14.5.2 下一步	504		

16.4.3	注册会话	549	附录 A	建立一个艺术图库	614
16.4.4	在会话里保存数组	551	A.1	项目概述	614
16.4.5	会话 Cookie 函数和配置 选项	554	A.2	公共和私有	614
16.4.6	利用会话设置偏爱	555	A.3	创建 Web 站点	615
16.4.7	命名会话	560	A.3.1	创建数据库	615
16.4.8	没有 Cookie 的会话	562	A.3.2	管理页面	616
16.4.9	利用链接传递会话 ID	565	A.3.3	公共页面	625
16.4.10	修改会话 ID	569	A.3.4	利用登录保护页面	631
16.4.11	终止会话	570	A.4	安装艺术画廊站点	632
16.4.12	会话运行时间配置	572	A.4.1	Canvas Gallery 站点的文件 在哪里	632
16.4.13	利用会话实现一个登录 系统	573	A.4.2	安装 MySQL 数据库	632
16.5	本章小结	577	A.4.3	编辑 PHP 页面	634
16.5.1	应该掌握的内容	577	A.5	小结	634
16.5.2	下一步	578	附录 B	PHP 与电子邮件	635
第 17 章	对象	579	B.1	邮件服务程序	635
17.1	什么是对象?	579	B.2	MIME (多用途互联网邮件 扩展)	636
17.2	使用类	580	B.3	运行时间配置选项	636
17.2.1	定义类	580	B.4	mail () 函数	636
17.2.2	类的实例化	581	B.5	发送一个简单的电子邮件消息	637
17.2.3	创建一个完整的类	584	B.6	范例: 发送一个 HTML 消息	638
17.2.4	显示一个对象	586	附录 C	PHP 与日期/时间	643
17.2.5	获取类信息的函数	587	C.1	格式化日期和时间	643
17.2.6	封装和信息隐藏	587	C.1.1	date() 函数	643
17.2.7	类成员和作用域	588	C.1.2	strftime() 函数	645
17.2.8	特殊方法	590	C.2	获取时标	647
17.2.9	继承	598	C.2.1	time() 函数	647
17.2.10	覆盖方法	601	C.2.2	mktime() 函数	648
17.2.11	访问受保护的成员	602	C.2.3	从字符串创建 UNIX 时标	650
17.3	PHP 5 的一些对象特性	605	C.3	获取日期和时间	651
17.3.1	final 类和方法	605	附录 D	安全与调试	655
17.3.2	静态成员	606	D.1	关于安全	655
17.3.3	类常数	607			
17.3.4	类的重用	608			
17.4	本章小结	612			

D.1.1	网络安全与 SSL	655	E.1	关于 Web 服务程序	667
D.1.2	操作系统、Web 服务程序 和文件系统的安全	656	E.2	在 Windows 上安装 Apache	667
D.2	保护 PHP 和 MySQL	656	E.3	在 Windows 上安装 PHP	668
D.3	调试	661	E.4	在 Linux/UNIX 上安装 PHP	669
D.3.1	启动错误报告	661	E.5	在 Mac OS X 上安装 PHP	669
D.3.2	首先纠正解析错误	662	E.6	配置 Apache 使用 PHP 模块 (各种平台)	669
D.3.3	用于调试的显示语句	662	E.7	配置 php.ini (各种平台)	670
D.3.4	纠正 SQL 错误	665	E.8	在 Windows 上安装 MySQL	671
D.3.5	更多调试技巧	666	E.9	在 Linux/UNIX 上安装 MySQL	671
附录 E	安装过程	667	E.10	在 Mac OS X 上安装 MySQL	672
			E.11	阅读手册	672

chapter

1

第 1 章 简介

1.1 从静态到动态站点

1.1.1 静态 Web 站点

“Web 实际上是我们通过共享信息来进行通信的公共信息空间……”

—— Tim Berners-Lee



Sir Tim Berners-Lee

Tim Berners-Lee 在 1989 年发明了万维网，于是引发了自 15 世纪古登堡发明印刷机之后的一次信息革命。不到十年时间，我们所生活的世界因此发生了根本的变化。

作为一名计算机顾问，当 Tim Berners-Lee 在位于瑞士日内瓦的物理实验室 CERN 工作时，迈开了他的革命步伐。CERN 是个庞大的科学研究中心，拥有数千名研究人员和数百个不同的系统。Tim Berners-Lee 最初尝试利用链接来管理自己硬盘上的文档，但最终形成了一种超文本语言，从而不仅能够从本机上，还能够从网络上的其他计算机来链接和分布相关联的文档。他的系统能够记录研究人员、他们的项目和论文、他们使用的软件、他们的计算机等。为了检索和发

送文档，他开发了一种简单的协议——HTTP（超文本传输协议）以及 HTML（超文本标记语言），用来描述文档中文本的布局。早期的 Web 很像在线图书馆：文档由链接相互连接，利用这些链接，我们就可以访问整个公司以至于整个世界的信息。

最初的 Web 是由政府投资的，只限于研究与教育领域。Web 站点包含由 HTML 语言编写的文档，页面都是基于文档的，是静态的。当用户在浏览器上重新加载页面时，会看到完全相同的

内容。页面上一般包含 HTML 文本、图像和链接。很显然，这是比较简单的，不像当今复杂的商业 Web 似乎可以做任何事情，比如在线购物、进行股票交易、预订休假计划，甚至是找对象。静态页面可以很好地发送和接收报告、图片和文章，但不能管理会改变的数据，不能记住用户的名称与偏好，不能根据数据库即时创建自定义的输出，或是随时在页面里嵌入流视频。随着 Web 不断成长，它演化为一个虚拟商场，商家需要 Web 站点通过交互和有趣的体验、快速响应与即时反馈来吸引潜在的购买者。这就是说，需要动态页面。

1.1.2 动态 Web 站点

动态站点的内容会在用户每次访问或重载站点时重新生成。其最简单的形式就是显示当前日期和时间，但在大多数实际应用中，这需要使用数据库。数据库里保存着站点的信息，而脚本语言可以从数据库里检索信息。比如 Google 和 Yahoo! 就是动态站点的很好范例，搜索引擎会根据用户输入的单词或短语生成不同的页面，这些页面是实时创建的，是专门根据用户输入而生成的。遍布全球的强大服务器会不断地接收这些请求，并且对其进行处理。在 Web 早期，处理过程是由“公共网关接口 (CGI)”完成的，这是一种服务器端技术，可以创建动态站点。大部分 CGI 脚本是用 Perl 编写的。浏览器把来自 HTML 页面的信息（比如来自表单的信息），发送到服务器进行处理。然后服务器就会创建一个通道，指向被称为 CGI 脚本或辅助程序的外部程序。虽然从理论上讲可以使用任何的编程语言，但编写 CGI 的最流行语言是 Perl。Perl 脚本编写的程序会分析数据、根据一定条件生成 HTML、发送电子邮件、打开文件或数据库，并且利用通道把信息发送回服务器，后者会把内容再传递给浏览器（参见图 1.1 的上半部分）。

虽然创建动态页面的底层过程并没有改变，但更新的编程语言出现了，极大简化了这个过程，使之能够嵌入到服务器里¹。PHP 就是这样一种语言。PHP 脚本能够内嵌在 Web 页面里，能够即时生成 HTML 和图像，从文件或数据库检索最新信息，加密数据，记住用户的偏好等。在执行 PHP 指令之后，它可以在服务器把页面发送给浏览器之前把结果插入到页面里，从而实现真正的动态页面（参见图 1.1 的下半部分）。

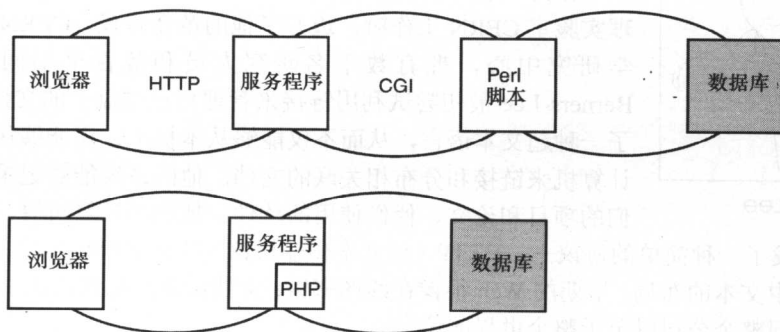


图 1.1 创建动态页面的过程

¹ 这里没有提及 Perl，因为它已经是过时的。Perl 具有 Mason 和 mod_perl，可以把 Perl 和 HTML 嵌入到 Apache 服务器。

Web 站点通常会处理大量信息。对于存储、检索和更新信息来说，数据库管理系统是基本需求。作为世界上最流行的开源数据库，MySQL 经常被用于需要数据库交互的 Web 站点。PHP 与 MySQL 的完美结合，形成了两种强大技术的强强联手。本书将展现如何实现这种结合。

1.1.3 什么是开放源码

“Free software (自由软件)”的意义在于自由权利，而不是价格。为了正确地理解这个概念，“free”应该理解为“free speech (自由说话的权利)”，而不是“free beer (免费啤酒)”

—— 自由软件协会

PHP 和 MySQL 代表着最新一代的开放源码程序。这意味着什么呢？Berners-Lee 最初的设想是让每个人都能自由地访问信息。但随着 Web 不断发展，人们对“free”的概念产生了不同的看法。但无论“free”是如何定义的，说专有软件¹（由版权所有人拥有和控制）不是免费的一般不会有什么问题。根据开放源码的初创精神，其目的在于让人们能够在很少的限制之下自由获得软件的源代码。

开放源码的基本观念是很简单的：当程序员能够读取、重新发布和修改软件的部分源代码时，这个软件就会成长。人们会增强它、修改它、改正其中的错误。对于习惯于传统软件开发步骤的人来说，这种改变的速度是非常令人惊讶的。

完整的讨论请见 <http://www.opensource.org/docs/definition.php>。

PHP 和 MySQL 都是开放源码的软件，简单来说，也就是不需要任何费用或试用期，我们就能够下载和使用这些程序。

1.2 关于 PHP



Rasmus Lerdorf

那么 PHP 是什么呢？PHP 是一种简单、快速可移植的脚本语言，非常适合开发具有数据库功能的 Web 站点。PHP 开发于 1995 年，目前已经用于全世界数以千万计的 Web 站点。PHP 的前身是 PHP/FI（个人家庭页面/表单解释器），这是由 Rasmus Lerdorf 在 1995 年开发的，用于追踪访问其在线履历的用户数。它基本上就是一组 Perl/CGI 脚本，稍后 Lerdorf 用 C 语言重写了它们，并且公布了源代码，也就是说可以自由获得。PHP 在语法上与 Perl 十分类似，但 Perl 是个功能完整、面向各种环境的脚本语言，而 PHP 是专门为控制 Web 而设计的。PHP 指令可以与 HTML 一起嵌入在 Web 页面里，这样无论何时页面被加载，PHP 都能够

¹ 私有软件的例子有 Microsoft Windows、Adobe Photoshop 和 WinZip。

执行它的代码。PHP 能够对表单变量进行自动解释，从而简化了表单处理过程。它能够与数据库进行交互，甚至能够让用户创建简单的动态 Web 站点。Rasmus Lerdorf 开发的工具集非常流行，所以在 1997 年，PHP/FI 2.0 出现了。由于这个新版本的广泛应用，Lerdorf 很快被吸纳到一个核心开发组，继续对这个新语言进行改善和增强。这时，有数千名用户和大约 50 000 个 Web 站点运行 PHP/FI 页面。

Zeev Suraski 和 Andi Gutmans 是以色列 Technion 技术学院的两个学生，在需要一种语言来完成大学电子商务项目时，他们选择了 PHP/FI。出于对其局限性与错误的不满，他们先把项目放到了一边，几乎是从头把 PHP 又重写了一遍。PHP 3.0 与前身代码有重大偏离，这种新语言支持附加模块，并且具有更一致的语法。与此同时，字母缩写的含义也改变了，PHP 现在代表“超文本预处理器”。PHP 3.0 是在 1998 年发布的，与当今使用的版本很类似。

2000 年 5 月，PHP 4 问世了。它的核心被彻底重写，从而改善了复杂 Web 应用的性能和提高了平台的模块化程度。作为 PHP 3 的作者，Zeev Suraski 和 Andi Gutmans 发明了一种新的解析引擎，名为 Zend 引擎¹，它正是支撑当今 PHP 的脚本语言。在得到了国际社会的广泛认可之后，Suraski 和 Gutmans 创建了 Zend 科技公司，也就是 PHP 公司，他们对 PHP 的促进是其应用范围呈爆炸性增长的主要原因。参见 www.zend.com。

PHP 4 提供了一个开放的“应用程序接口 (API)”，从而允许其他程序员为 PHP 编写模块。这些模块能够扩展 PHP 的功能，能够让 PHP 4 支持大多数流行的数据库和 Web 服务器。在发布了这个版本之后，PHP 成为开发与部署复杂 Web 应用的正式编程语言和平台。

PHP 最新的版本是 5.0，发布于 2004 年 7 月，它添加了全新的面向对象模块。这个新模块基于 Zend 2 引擎，极大地提高了 PHP 的性能和功能。其中大多数功能是向后兼容的，允许老版本编写的程序能够继续工作。

根据 Netcraft 的统计数据，截止到 2005 年 10 月，有 23 299 550 个域和 1 290 179 个 IP 地址支持使用 PHP。参见 <http://www.php.net/usage.php>。

在哪里获得 PHP 和文档

在 PHP 的官方站点 php.net (参见图 1.2) 可以获得用于 Apache 和 Microsoft 的 PHP 服务程序的最新版本。这个站点还提供了最新的 PHP 文档；我们可以在页面右上角输入搜索字符串，返回的结果不仅包含特定的函数，还有链接指向其他能够完成类似任务的函数。大多数官方文档页面都包含来自其他用户的注释、错误修正与版本修订 (参见图 1.3)。

¹ 术语“Zend”是个合成词，是由他们名字 Zeev 和 Andrew 的前两个字母组成的。