

普.通.高.等.学.校
计算机教育“十二五”规划教材

PHP 编程基础 与实例教程

(第2版)

PHP FUNDAMENTALS & PRACTICES
(2nd edition)

孔祥盛 ◆ 主编



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

普.通.高.等.学.校
计算机教育“十二五”规划教材

PHP 编程基础 与实例教程

(第2版)

PHP FUNDAMENTALS & PRACTICES
(2nd edition)

孔祥盛 ◆ 主编

人民邮电出版社
北京

图书在版编目 (CIP) 数据

PHP编程基础与实例教程 / 孔祥盛主编. -- 2版. --
北京: 人民邮电出版社, 2016.6 (2016.7重印)
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-42055-8

I. ①P… II. ①孔… 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(2016)第061121号

内 容 提 要

PHP 简单易学且功能强大, 是开发 Web 应用程序理想的脚本语言。本书由浅入深、循序渐进, 系统地介绍了 PHP 的相关知识及其在 Web 应用程序开发中的实际应用, 并通过具体案例, 使读者巩固所学知识, 更好地进行开发实践。本书共分为 13 章, 涵盖了 PHP 开发环境的搭建、PHP 语法、FORM 表单、数据库设计、MySQL 数据库、PHP 与数据库连接、会话控制、界面设计等内容。

本书内容丰富、讲解深入, 适用于初、中级 PHP 用户, 可以作为各类院校相关专业的教材, 同时也是一本面向广大 PHP 爱好者的 PHP 实用参考书。

-
- ◆ 主 编 孔祥盛
责任编辑 邹文波
责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 23.75 2016 年 6 月第 2 版
字数: 625 千字 2016 年 7 月河北第 2 次印刷
-

定价: 54.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

5. 丰富而实用的课后习题：精选新浪、百度等知名公司面试题。

本书由孔祥盛任主编，张永华、张元好、茹蓓、李彦、李军伟任副主编。其中，孔祥盛编写第 1、7、9、13 章，设计了本书案例、组织架构，并进行全书统稿；李彦编写第 2、3、4 章；李军伟编写第 5 章和第 6 章；张元好编写第 8 章；茹蓓编写第 10 章和第 11 章；张永华编写第 12 章。此外，任卫银负责了本书案例的界面设计，孙大鹏、李敏、李辉、王珍、侯国平、赵春霞、王娜负责了本书的代码测试。

未经许可，不得以任何方式复制或抄袭本书部分或全部内容。版权所有，侵权必究。

编 者

2016 年 3 月

目 录

第 1 章 PHP 入门1	
1.1 PHP 概况.....1	
1.1.1 PHP 的优势.....1	
1.1.2 PHP 的应用领域.....2	
1.1.3 HTML 内嵌式的脚本语言.....2	
1.2 PHP 脚本程序工作流程.....3	
1.2.1 Web 浏览器.....3	
1.2.2 HTML 代码.....3	
1.2.3 PHP 预处理器.....3	
1.2.4 Web 服务器.....3	
1.2.5 HTTP 协议.....3	
1.2.6 数据库服务器.....4	
1.2.7 PHP 程序的工作流程.....5	
1.3 PHP 服务器的构建.....6	
1.3.1 服务器安装前的准备工作.....6	
1.3.2 安装服务器.....8	
1.3.3 启动与停止服务.....10	
1.3.4 第一个 PHP 程序.....12	
1.3.5 配置服务器.....12	
习题.....17	
第 2 章 PHP 基础18	
2.1 PHP 代码基本语法.....18	
2.1.1 PHP 开始标记与结束标记.....18	
2.1.2 PHP 注释.....20	
2.1.3 PHP 语句及语句块.....21	
2.2 PHP 程序的组成.....22	
2.2.1 关于 PHP 数据.....23	
2.2.2 PHP 数据类型.....23	
2.2.3 浏览器端的数据采集.....28	
2.2.4 PHP 程序的数据采集.....29	
2.2.5 PHP 数据处理.....29	
2.2.6 PHP 数据的输出.....30	
2.3 编码规范.....32	
2.3.1 书写规范.....32	
2.3.2 命名规范.....33	
2.3.3 为代码添加注释.....34	
习题.....34	
第 3 章 PHP 表达式35	
3.1 常量.....35	
3.1.1 自定义常量.....35	
3.1.2 常量的内存分配.....36	
3.1.3 预定义常量.....36	
3.2 变量.....37	
3.2.1 变量的基本概念.....37	
3.2.2 变量的内存分配.....38	
3.2.3 变量的赋值方式.....39	
3.2.4 可变变量.....41	
3.3 有关变量或常量状态的函数.....41	
3.3.1 数据类型查看函数.....41	
3.3.2 检查常量或变量是否定义函数.....42	
3.3.3 取消变量定义 unset()函数.....44	
3.3.4 检查变量是否为“空”.....45	
3.3.5 数据类型检查函数.....48	
3.4 PHP 运算符.....49	
3.4.1 算术运算符.....49	
3.4.2 递增/递减运算符.....49	
3.4.3 赋值运算符.....50	
3.4.4 比较运算符.....51	
3.4.5 逻辑运算符.....51	
3.4.6 字符串连接运算符.....52	
3.4.7 错误抑制运算符.....53	
3.4.8 条件运算符.....53	
3.4.9 类型运算符.....54	
3.4.10 执行运算符.....54	
3.4.11 位运算符.....54	
3.4.12 运算符优先级.....55	

3.5 数据类型的转换	55	5.4.8 数组与数据结构	106
3.5.1 类型自动转换	56	5.4.9 数组集合运算函数	107
3.5.2 类型强制转换	58	习题	109
习题	60	第 6 章 PHP 的数据采集	113
第 4 章 PHP 流程控制语句	65	6.1 浏览器端数据的提交方式	113
4.1 条件控制结构	65	6.1.1 GET 提交方式	113
4.1.1 if 语句	65	6.1.2 POST 提交方式	115
4.1.2 if...else 语句	66	6.1.3 GET 和 POST 混合提交方式	116
4.1.3 switch 语句	68	6.1.4 两种提交方式的比较	116
4.2 循环结构	70	6.2 相对路径和绝对路径	117
4.2.1 while 循环语句	70	6.2.1 绝对路径	117
4.2.2 do...while 循环语句	70	6.2.2 相对路径	117
4.2.3 for 循环语句	71	6.2.3 相对路径其他概念	118
4.2.4 循环结构应用	72	6.3 使用 FORM 表单实现浏览器端的	
4.3 程序跳转和终止语句	73	数据采集	118
4.3.1 continue 语句	73	6.3.1 表单标签 <form></form>	119
4.3.2 break 语句	74	6.3.2 表单控件	119
4.3.3 终止 PHP 程序运行	74	6.3.3 在表单控件中使用数组	123
习题	75	6.3.4 表单按钮	123
第 5 章 PHP 数组	77	6.3.5 FORM 表单综合应用	125
5.1 数组的基本概念	77	6.4 使用\$_GET 和\$_POST “采集”	
5.1.1 为什么引入数组	77	表单数据	127
5.1.2 数组的分类	78	6.5 上传文件的 “数据采集”	128
5.2 一维数组的创建	78	6.5.1 与上传相关的配置	128
5.2.1 直接将变量声明为数组元素	78	6.5.2 PHP 文件上传流程	129
5.2.2 使用 array() 语言结构创建数组	80	6.5.3 预定义变量\$_FILES	131
5.2.3 创建数组的注意事项	80	6.5.4 PHP 文件上传的实现	131
5.2.4 数组元素 “值” 的访问	81	6.6 Web 服务器端其他数据采集方法	133
5.3 二维数组	83	6.6.1 预定义变量\$_REQUEST	133
5.3.1 二维数组的创建	84	6.6.2 预定义变量\$_SERVER	134
5.3.2 二维数组元素 “值” 的访问	85	习题	135
5.4 数组处理函数及应用	86	第 7 章 自定义函数	137
5.4.1 快速创建数组的函数	86	7.1 PHP 文件间的引用	137
5.4.2 数组统计函数	88	7.1.1 include 语言结构	137
5.4.3 数组指针函数	91	7.1.2 require 与 include 语言结构的	
5.4.4 数组和变量间的转换函数	96	比较	138
5.4.5 数组遍历语言结构	98	7.1.3 include_once 和 require_once	
5.4.6 数组检索函数	101	语言结构	139
5.4.7 数组排序函数	103	7.2 函数概述	140

7.2.1 函数的概念	140	8.5.4 删除数据库	178
7.2.2 函数的分类	140	8.6 数据库表的管理	178
7.3 自定义函数	140	8.6.1 数据类型	178
7.3.1 自定义函数的定义	141	8.6.2 MySQL 中的附加属性	179
7.3.2 自定义函数的声明和调用	143	8.6.3 创建数据库表	180
7.3.3 自定义函数的参数赋值	143	8.6.4 显示数据库表结构	182
7.3.4 变量的作用域和 global 关键字	147	8.6.5 删除数据库表结构	183
7.3.5 变量的生存周期和 static 关键字	149	8.7 表记录的更新操作	183
7.3.6 变量函数	152	8.7.1 表记录的添加	183
7.4 自定义函数综合示例	153	8.7.2 表记录的修改	185
7.5 return 语言结构	154	8.7.3 表记录的删除	186
习题	156	8.8 表记录的查询操作	186
第 8 章 MySQL 数据库	160	8.8.1 指定字段列表及列别名	187
8.1 数据库概述	160	8.8.2 使用谓词限制记录的行数	187
8.1.1 数据库	160	8.8.3 使用 from 子句指定多个数据源	188
8.1.2 关系数据库管理系统 (RDBMS)	161	8.8.4 使用 where 子句过滤记录	191
8.1.3 结构化查询语言 (SQL)	161	8.8.5 使用 order by 子句对记录排序	193
8.2 数据库规范化设计	161	8.8.6 使用聚合函数返回汇总值	194
8.2.1 E-R 模型	162	8.8.7 使用 group by 子句对记录分组 统计	194
8.2.2 主键 (Primary Key)	164	8.8.8 使用 having 子句提取符合条件 的分组	194
8.2.3 实体间的关系与外键 (Foreign Key)	165	8.9 MySQL 特殊字符序列	195
8.2.4 约束 (Constraint)	166	习题	196
8.3 MySQL 简介	167	第 9 章 PHP 与 MySQL 数据库	200
8.3.1 MySQL 服务的启动	167	9.1 PHP 中常用的 MySQL 操作函数	200
8.3.2 MySQL 客户机	167	9.1.1 连接 MySQL 服务器	200
8.3.3 连接 MySQL 服务器	169	9.1.2 设置数据库字符集	201
8.3.4 MyISAM 和 InnoDB 存储引擎	170	9.1.3 关闭 MySQL 服务器连接	201
8.4 字符集	172	9.1.4 选择当前操作的数据库	201
8.4.1 字符集与字符序	173	9.1.5 发送 SQL 语句或 MySQL 命令	202
8.4.2 MySQL 字符集与字符序	173	9.1.6 遍历结果集中的数据	204
8.4.3 MySQL 的字符集转换过程	175	9.1.7 MySQL 服务器连接与关闭最佳 时机	206
8.4.4 设置 MySQL 字符集	176	9.1.8 MySQL 服务器连接与关闭函数的 制作	207
8.4.5 SQL 脚本文件	176	9.2 PHP 中其他 MySQL 操作函数	207
8.5 MySQL 数据库管理	177	9.2.1 数据库表操作函数	207
8.5.1 创建数据库	177	9.2.2 选择当前操作的数据库并发送	
8.5.2 选择当前操作的数据库	177		
8.5.3 显示数据库结构	178		

SQL 语句	208	实施	235
9.2.3 表字段操作函数	209	10.6 分页原理及实现	246
9.2.4 其他常用函数	210	10.6.1 分页原理	246
9.3 用户注册系统的实现	212	10.6.2 PHP 分页的最简单实现	247
9.3.1 用户注册系统文件组织结构	212	10.6.3 带有“分页导航条”分页的 实现	247
9.3.2 用户注册界面的实现	212	10.6.4 显示分页导航条的函数制作	249
9.3.3 数据库的实现	214	10.7 新闻发布系统的软件测试	251
9.3.4 制作用户注册系统所需的函数	214	10.8 新闻发布系统的系统支持	251
9.3.5 用户注册功能的实现	214	习题	252
9.3.6 用户登录页面的实现	217	第 11 章 PHP 会话控制	253
9.3.7 用户登录功能的简单实现	217	11.1 HTTP 无状态特性	253
9.3.8 功能测试	218	11.1.1 HTTP 通信机制	253
9.4 SQL 注入	219	11.1.2 HTTP 无状态与 TCP 长连接 之间的关系	255
习题	220	11.1.3 HTTP 请求信息	255
第 10 章 新闻发布系统的开发	222	11.1.4 HTTP 响应信息	257
10.1 新闻发布系统的开发流程	222	11.2 页面间的参数传递	258
10.1.1 MIS 的开发流程	222	11.2.1 利用重定向实现参数传递	259
10.1.2 新闻发布系统的开发流程	222	11.2.2 使用 JavaScript 实现重定向	260
10.2 新闻发布系统的系统规划	222	11.2.3 使用 PHP 实现重定向	260
10.2.1 新闻发布系统的目标	223	11.3 Cookie 会话技术	261
10.2.2 新闻发布系统的可行性分析	223	11.3.1 浏览器的 Cookie 设置	261
10.2.3 新闻发布系统的项目进度表	223	11.3.2 Cookie 的工作原理	261
10.2.4 新闻发布系统的人员分工	223	11.3.3 Cookie 的内容	262
10.3 新闻发布系统的系统分析	223	11.3.4 Cookie 分类及典型应用	263
10.3.1 新闻发布系统的功能需求	224	11.3.5 使用 Cookie 的步骤	264
10.3.2 新闻发布系统的技术要求	225	11.3.6 创建 Cookie	265
10.3.3 新闻发布系统中使用的模型	226	11.3.7 预定义变量\$_COOKIE	266
10.3.4 新闻发布系统的 E-R 模型	226	11.3.8 删除浏览器端的 Cookie	268
10.3.5 新闻发布系统的数据流程图	227	11.3.9 新闻发布系统用户管理功能的 实现 (一)	268
10.3.6 数据字典 (Data Dictionary)	230	11.3.10 Cookie 数组的使用	271
10.4 新闻发布系统的系统设计	231	11.3.11 使用 Cookie 的其他注意事项	272
10.4.1 系统流程图	231	11.4 Session 会话技术	273
10.4.2 程序流程图	232	11.4.1 Session 的工作原理	273
10.4.3 数据库规范化设计	233	11.4.2 使用 Session 的步骤	274
10.4.4 图形用户界面设计	233	11.4.3 php.ini 有关 Session 的配置	276
10.5 新闻发布系统系统实施	234	11.4.4 开启 Session	276
10.5.1 文件组织结构	234		
10.5.2 数据库的实施	235		
10.5.3 新闻管理和评论管理功能的			

11.4.5 预定义变量\$_SESSION	277	12.3.3 修改发表评论功能的代码	336
11.4.6 删除和销毁 Session	278	12.3.4 优化新闻列表显示功能的代码	337
11.4.7 Session 的综合应用	278	12.3.5 模糊查询时关键字以加粗倾斜 格式显示	337
11.4.8 Session ID、Session name 和 SID	280	12.3.6 优化分页函数代码	338
11.4.9 禁用 Cookie 后 Session 的实现	283	习题	339
11.4.10 Session 和 Cookie 的对比	284	第 13 章 新闻发布系统的页面 美工	343
11.4.11 新闻发布系统用户管理功能的 实现(二)	285	13.1 JavaScript 脚本语言	343
11.4.12 新闻发布系统权限控制的实现	288	13.2 FCKeditor 在线编辑器	344
11.4.13 使用 Session 数组模拟购物车 功能	290	13.2.1 FCKeditor 使用前的准备工作	344
11.5 header()函数的使用	291	13.2.2 FCKeditor 类的成员变量和成员 方法	345
11.5.1 页面重定向	292	13.2.3 FCKeditor 的高级配置	345
11.5.2 创建 Cookie	293	13.2.4 FCKeditor 在新闻发布系统中的 应用	347
11.5.3 服务器响应内容的控制	293	13.2.5 FCKeditor 的文件管理	349
11.5.4 完善新闻发布系统文件下载功能	297	13.2.6 FCKeditor 瘦身	351
11.5.5 浏览器缓存的远程控制	299	13.3 新闻发布系统页面布局的实现	351
11.5.6 常用的浏览器缓存控制函数	303	13.3.1 DIV+CSS 概述	352
习题	305	13.3.2 界面布局图	352
第 12 章 字符串处理	307	13.3.3 使用 DIV 实现页面布局	353
12.1 字符串的指定方法	307	13.3.4 准备图片素材	354
12.1.1 使用单引号指定字符串	307	13.3.5 CSS 热身	354
12.1.2 使用双引号指定字符串	307	13.3.6 header 层的实现	357
12.1.3 使用定界符指定字符串	309	13.3.7 pagebody 层的实现	360
12.1.4 字符串中的字符处理	310	13.3.8 footer 层的实现	360
12.2 字符串处理函数	310	13.4 新闻发布系统静态和动态页面的 嵌入	361
12.2.1 字符串修剪函数	310	13.4.1 将用户管理功能嵌入到 login 层	361
12.2.2 字符串长度函数	320	13.4.2 修改 menu 层代码	361
12.2.3 子字符串操作函数	322	13.4.3 将主要功能嵌入到 mainfunction 层	363
12.2.4 字符串比较函数	326	13.4.4 sidebar 层和 mainbody 层的高度 自适应功能	367
12.2.5 字符串连接和分割函数	327	13.4.5 防止图片太宽撑破 mainbody 层	369
12.2.6 字符串替换函数	328	习题	369
12.2.7 URL 处理函数	329	参考文献	370
12.2.8 其他常用的字符串函数	334		
12.3 新闻发布系统中的字符串处理函数 的应用	335		
12.3.1 删除模糊查询中关键字两边的 空格	336		
12.3.2 修改文件下载功能的代码	336		

第 1 章

PHP 入门

本章首先介绍 PHP 概况，然后介绍 PHP 程序的工作流程，并以 WampServer 为例介绍 PHP 服务器的安装和配置。通过本章的学习，读者可以了解 PHP 程序的工作流程，并可以编写、运行简单的 PHP 程序。

1.1 PHP 概况

PHP 是 Hypertext Preprocessor 的缩写，是一种被广泛应用的、免费开源的、服务器端的、跨平台的、HTML 内嵌式的多用途脚本语言。PHP 通常嵌入到 HTML 中，尤其适合 Web 开发。PHP 与微软公司的 ASP（或 NET）以及甲骨文 Oracle 公司的 JSP 颇有几分相似，是一种在服务器端执行的 HTML 内嵌式的脚本语言。

1.1.1 PHP 的优势

PHP 发展到今天，具备了很多优势，简单介绍如下。

1. 易学好用：学习 PHP 的过程非常简单。PHP 的主要目标是让 Web 开发人员只需很少的编程知识就可以快速地建立一个真正动态交互的 Web 系统。PHP 语言的风格类似于 C 语言，非常容易学习，只要了解一点儿 PHP 的基本语法和语言特色，就可以开始 PHP 编程之旅。

2. 免费开源：基于 PHP 的 Web 系统源代码是免费开源的。

3. 良好的可扩展性：PHP 的免费开源导致可扩展性大大增强，任何程序员为 PHP 扩展附加功能都非常容易。

4. 平台无关性（跨平台）：同一个 PHP 应用程序，无需修改任何源代码，就可以运行在 Windows、Linux、UNIX 等绝大多数操作系统环境中。

5. 功能全面：PHP 几乎涵盖了 Web 系统所需的一切功能，例如，使用 PHP 可以进行图形处理、编码与解码、压缩文件处理、XML 解析、支持 HTTP 的身份认证、Session 和 Cookie 等操作。

6. 数据库支持：PHP 最强大、最显著的优势是支持包括甲骨文公司的 Oracle 及 MySQL、微软公司的 Access 及 SQL Server 在内的大部分数据库管理系统，并且使用 PHP 编写数据库支持的动态网页非常简单。

7. 面向对象编程：PHP 较新版本提供了面向对象的编程方式，不仅提高了代码的重用率，而且为代码维护带来很大的方便。

1.1.2 PHP 的应用领域

PHP 主要应用于以下 3 个领域。

1. 服务器端脚本程序：可以使用 PHP 编写服务器端的脚本程序，完成任何其他的脚本语言（例如 ASP、JSP 或 .NET）完成的工作，例如，收集表单数据，生成动态网页，或者发送/接收 Cookie 等。

2. 命令行脚本程序：可以使用 PHP 编写一段命令行脚本程序。运行命令行脚本程序时，只需借助 PHP 预处理器，无需借助任何 Web 服务器和 Web 浏览器。

3. 桌面应用程序：可以使用 PHP 编写图形界面的桌面应用程序。当然对于桌面应用程序而言，PHP 并不是最好的选择。

使用 PHP 编写服务器端脚本程序是 PHP 最常用的应用领域，这也是本书着重阐述的内容。

1.1.3 HTML 内嵌式的脚本语言

PHP 脚本程序中可包含文本、HTML 代码以及 PHP 代码。例如，程序 helloworld.php 如下。

这是我的第一个 PHP 程序：

```
<br/>
<?php
echo "hello world!";
?>
<br/>
<?php
echo date("Y年m月d日H时i分s秒");
?>
```

程序 helloworld.php 中，各部分说明如下。

1. “这是我的第一个 PHP 程序：” 是一段文本信息。PHP 程序中的文本信息将不被 PHP 预处理器处理，直接被 Web 服务器输出到 Web 浏览器。

2. “
” 是 HTML 代码。PHP 程序中的 HTML 代码同样不被 PHP 预处理器处理，直接被 Web 服务器输出到 Web 浏览器，只不过 Web 浏览器接收到 HTML 代码后，会对该 HTML 代码解释执行，例如 Web 浏览器接收到 “
” 后，将在 Web 浏览器产生一次换行。

3. “echo "hello world!";” 和 “echo date("Y年m月d日H时i分s秒");” 是两条 PHP 代码，所有的 PHP 代码都要经 PHP 预处理器解释执行。PHP 预处理器解释这两条 PHP 代码时，会将这两条代码解释为文本信息 “hello world!” 和 Web 服务器主机的当前时间（如 “2015 年 8 月 11 日 13 时 41 分 31 秒”），然后再将这些文本信息输出到 Web 浏览器，最后由 Web 浏览器显示这些文本信息。

4. date() 是一个日期时间函数，该函数需要一个字符串参数，例如 “Y 年 m 月 d 日 H 时 i 分 s 秒”。Y 是 year 的第一个字母，m 是 month 的第一个字母，d 是 day 的第一个字母，H 是 hour 的第一个字母，i 是 minute 的第二个字母，s 是 second 的第一个字母，分别代表 Web 服务器当前的年、月、日、时、分、秒。



PHP 代码通常以符号 “<?php” 开始，符号 “?>” 结束，这两个符号分别叫做 PHP 开始标记和结束标记。

注意 PHP 程序文件名中的扩展名通常使用 “.php”，如 helloworld.php。

1.2 PHP 脚本程序工作流程

运行 PHP 脚本程序，必须借助 PHP 预处理器、Web 服务器和 Web 浏览器，必要时还需借助数据库服务器。其中 Web 服务器的功能是处理 HTTP 请求，PHP 预处理器的功能是解释 PHP 代码，Web 浏览器的功能是显示 PHP 程序的执行结果，数据库服务器的功能是存储执行结果。

1.2.1 Web 浏览器

Web 浏览器 (Web Browser) 也叫网页浏览器，简称浏览器。浏览器是用户最为常用的客户端程序，主要功能是显示 HTML 网页内容，并让用户与这些网页内容产生互动。常见的浏览器有微软公司的 Internet Explorer (简称 IE) 浏览器、Mozilla 公司的 Firefox 浏览器等。

1.2.2 HTML 代码

HTML 是 Hypertext Markup Language (超文本标记语言) 的缩写，HTML 代码是网页的静态内容，这些静态内容由 HTML 标记产生，Web 浏览器识别这些 HTML 标记并解释执行。例如 Web 浏览器识别 HTML 标记“
”，将“
”标记解析为一个换行。在 PHP 程序开发过程中，HTML 代码主要负责页面的互动、布局 and 美观。

1.2.3 PHP 预处理器

PHP 预处理器 (PHP Preprocessor) 的功能是将 PHP 程序中 PHP 代码解释为文本信息，这些文本信息中可以包含 HTML 代码。

1.2.4 Web 服务器

Web 服务器 (Web Server) 也称为 WWW (World Wide Web) 服务器，简单地说，安装有 Web 服务器软件的计算机称为 Web 服务器。常用的 Web 服务器软件有微软公司的 Internet Information Server (IIS) 服务器软件、IBM 公司的 WebSphere 服务器软件以及开源的 Apache 服务器软件等。由于 Apache 具有免费、速度快且性能稳定等特点，它已成为目前最为流行的 Web 服务器软件。本书将使用 Apache 服务器部署 PHP 程序。无论哪一种 Web 服务器，它们主要提供以下两个功能。

(1) 存储大量的网络资源以供浏览器用户访问。典型的网络资源包括静态页面、动态页面以及各种多媒体网络资源 (如图片、音频、视频、Flash 等资源)。



Web 服务器上的静态页面通常以“.html”或者“.htm”为文件扩展名；动态页面通常以“.php”为文件扩展名。

(2) 处理 HTTP 请求。

1.2.5 HTTP 协议

超文本传输协议 (HyperText Transfer Protocol, HTTP) 定义了 Web 浏览器与 Web 服务器通过网络进行无状态通信的一套规则。简单地说，无状态是指当一个 Web 浏览器向某个 Web 服务器的页面发送请求 (Request) 后，Web 服务器收到该请求进行处理，然后将处理结果作为响应 (Response) 返

回给 Web 浏览器，Web 浏览器与 Web 服务器都不保留当前 HTTP 通信的相关信息。也就是说，Web 浏览器打开 Web 服务器上的一个网页，和之前打开这个服务器上的另一个网页之间没有任何联系。

HTTP 遵循请求（Request）/响应（Response）模型，所有 HTTP 通信连接都被构造成一对外 HTTP 请求和 HTTP 响应。HTTP 请求类型多种多样，有以下几种分类方法。

1. 按照请求方法的不同，可将 HTTP 请求分为 GET 请求、POST 请求、HEAD 请求、OPTIONS 请求、PUT 请求、DELETE 请求和 TARCE 请求，其中最为常用的请求方法是 GET 请求和 POST 请求。本书将在 PHP 数据的采集章节对这两种请求方法进行详细讲解。

2. 按照请求的资源类型不同，可将 HTTP 请求分为 HTTP 动态请求及 HTTP 静态请求。当 Web 浏览器访问 Web 服务器上的静态页面时，此时的 HTTP 请求为静态请求；反之为动态请求。

大部分 Web 服务器仅提供一个可以执行服务器端程序和返回响应的环境，单纯的 Web 服务器只能响应静态页面（例如不包含任何 PHP 代码的 HTML 页面）的静态请求。也就是说，如果 Web 浏览器请求的是静态页面，此时只需要 Web 服务器响应该请求；如果浏览器请求的是动态页面（例如页面中包含了 PHP 代码），此时 Web 服务器会委托 PHP 预处理器将该动态页面的 PHP 代码解释为 HTML 静态页面，然后再将静态页面返回给浏览器进行显示。



1.2.6 数据库服务器

简单地说，数据库（Database，DB）是存储、管理数据的容器。数据库容器通常包含诸多数据库对象，如表、视图、索引、函数、存储过程、触发器等。这些数据库对象最终都是以文件的形式存储在外存（如硬盘）上。数据库用户如何访问数据库容器中的数据库对象呢？事实上，通过“数据库管理系统”数据库用户可以轻松地实现数据库容器中各种数据库对象的访问（增、删、改、查等操作），并可以轻松地完成数据库的维护工作（备份、恢复、修复等操作），如图 1-1 所示。

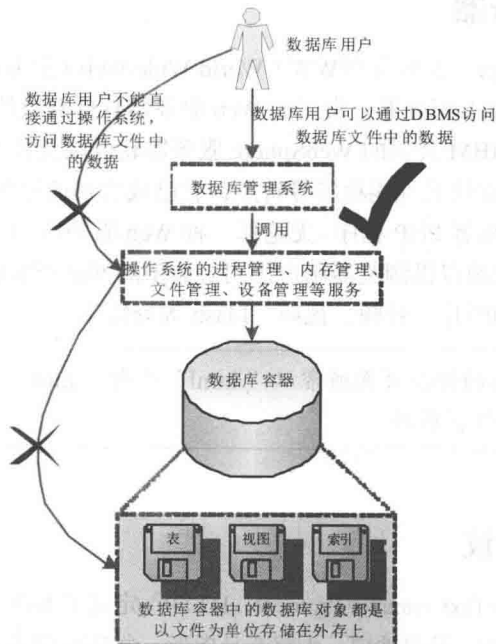


图 1-1 数据库管理系统与操作系统之间的关系

简单地说，安装有数据库管理系统软件的计算机称为数据库服务器（DataBase Server）。数据库管理系统（Database Management System, DBMS）安装于操作系统之上，是一个管理、控制数据库容器中各种数据库对象的系统软件。可以这样理解：数据库用户无法直接通过操作系统获取数据库文件中的具体内容；数据库管理系统通过调用操作系统的进程管理、内存管理、设备管理及文件管理等服务，为数据库用户提供管理、控制数据库容器中各种数据库对象、数据库文件的接口。

目前成熟的数据库管理系统主要源自欧美数据库厂商，典型的有美国甲骨文公司的 Oracle 和 MySQL、美国微软公司的 SQL Server、德国 SAP 公司的 Sybase 以及美国 IBM 公司的 DB2 和 Informix。这些数据库管理系统除了 MySQL 是开源数据库外，其他都是商业数据库，价格昂贵。考虑到 MySQL 开源、免费、易于安装、性能高效、功能齐全等特点，许多中小型 Web 系统选择 MySQL 作为首选数据库管理系统。本书也将选用 MySQL 详细讲解有关 PHP 数据库开发方面的知识。

1.2.7 PHP 程序的工作流程

PHP 程序的工作流程可以通过图 1-2 进行简单描述，具体步骤如下。

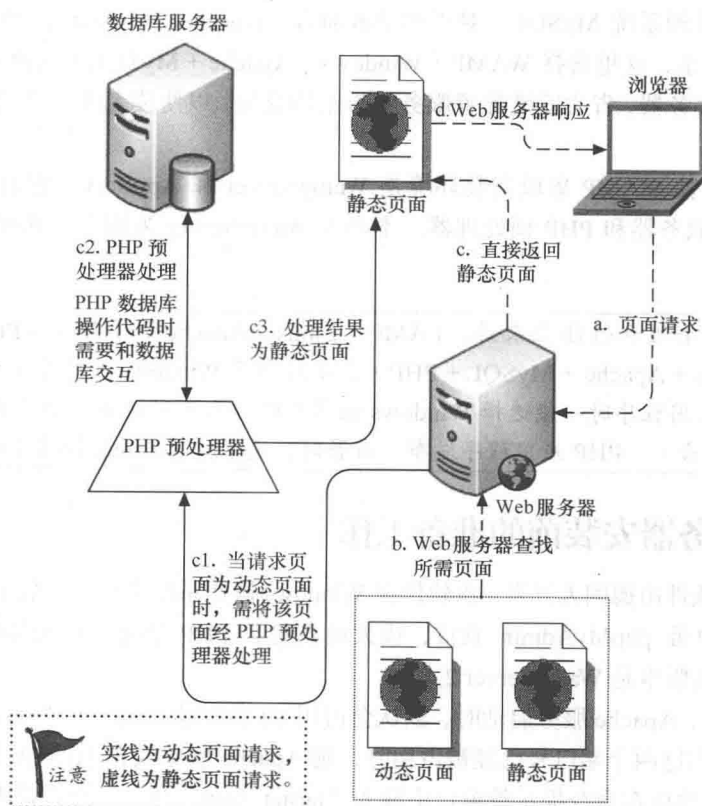


图 1-2 PHP 程序的工作流程

1. 用户在浏览器地址栏中输入要访问的页面地址（形如 `http://localhost/1/helloworld.php`），按回车键后就会触发该页面请求，并将请求传送给 Web 服务器（步骤 a）。
2. Web 服务器接收到该请求后，根据请求页面文件名在 Web 服务器主机中查找对应的页面文

件 (步骤 b)，并根据请求页面文件名的扩展名 (如.html 或.php) 判断当前 HTTP 请求为静态页面请求还是动态页面请求。

(1) 当请求页面为静态页面时 (如请求页面文件扩展名为.html 或.htm)，Web 服务器直接将请求页面返回 (步骤 c)，并将该页面作为响应发送给浏览器 (步骤 d)。

(2) 当请求页面为动态页面时 (如请求页面文件扩展名为.php)，此时 Web 服务器委托 PHP 预处理器将该动态页面中的 PHP 代码解释为文本信息 (步骤 c1)；如果动态页面中存在数据库操作代码，PHP 预处理器和数据库服务器完成信息交互 (步骤 c2) 后，再将动态页面解释为静态页面 (步骤 c3)；最后 Web 服务器将该静态页面作为响应发送给浏览器 (步骤 d)。

3. 浏览器接收到 Web 服务器的 HTTP 响应后，将执行结果显示在浏览器或由浏览器进行其他处理。

1.3 PHP 服务器的构建

为了构建 PHP 服务器，在服务器软件的选择上，我们选择免费开源的 Web 服务器软件 Apache 和数据库管理系统 MySQL。对于初学者而言，Apache、MySQL 以及 PHP 预处理器的安装和配置较为复杂，这里选择 WAMP (Windows + Apache + MySQL + PHP) 集成安装环境快速安装配置 PHP 服务器，省去安装配置服务器带来的麻烦，以便读者能够更快地进入 PHP 编程的殿堂。

目前常用的两款 WAMP 集成安装环境是 WampServer 和 AppServ，它们都集成了 Apache 服务器、MySQL 服务器和 PHP 预处理器。本书以 WampServer 为例介绍 PHP 服务器的安装和配置。



从安全性和性能上来讲，LAMP (Linux + Apache + MySQL + PHP) 优于 WAMP (Windows + Apache + MySQL + PHP)，不过由于 Windows 操作系统更易使用，因此开发 PHP 应用程序时一般选择 Windows 操作系统作为开发环境；由于 PHP 具有平台无关性 (跨平台)，PHP 应用程序发布、部署时，通常使用 Linux 操作系统。

1.3.1 服务器安装前的准备工作

WampServer 软件由德国人开发。该软件在 Windows 操作系统平台下集成了 Apache、MySQL 和 PHP，其中还自带 phpMyAdmin 软件，极大地方便了 PHP 服务器的安装配置和使用。目前 WampServer 的较新版本是 WampServer 2.4。

需要注意的是，Apache 服务启动时，默认会占用 80 端口号；MySQL 服务启动时，默认会占用 3306 端口号。当这两个端口号已经被占用时，则 Apache 服务或者 MySQL 服务会启动失败。在服务启动之前，建议在命令提示符窗口中输入 “netstat -aon” 命令，查看是否有进程占用了 80 和 3306 端口号；如果有，记住进程唯一标记符 PID，然后通过下列方法杀死该进程，以便释放 80 或者 3306 端口号，确保 Apache 服务及 MySQL 服务启动成功。

方法 1：记录对应进程的 PID，然后在 CMD 命令行窗口中执行 “taskkill PID” 命令，杀死该进程。

方法 2：首先，通过端口号找到进程 PID，命令格式：netstat -aon | findstr "端口号"。其次，

通过 PID 号找进程名，命令格式：`tasklist | findstr "PID"`。最后打开任务管理器杀死该进程名的进程。

本书为了区分服务、服务进程以及服务器，进行如下定义（以 Apache 为例）。

Apache 服务：实际上是保存在 Web 服务器硬盘上的一个服务软件，是静态的代码集合。

Apache 服务进程：如果将 Apache 服务软件加载到内存，并且正在占用 CPU 动态运行，此时就生成了 Apache 服务进程，只有处于运行状态的 Apache 服务进程才可以响应 Web 浏览器的请求。同一个 Apache 服务软件，如果 Apache 配置文件的参数不同，启动 Apache 服务后生成的 Apache 服务进程也不相同。



Web 服务器：实际上是一个安装有 Apache 服务的主机系统的总称，该主机系统还应该包括操作系统、CPU、内存及硬盘等软硬件资源。特殊情况下，同一台 Apache 服务器可以安装多个 Apache 服务，甚至可以同时运行多个 Apache 服务进程，各 Apache 服务进程占用不同的端口号为不同的 Web 浏览器提供服务。简而言之，同一台 Apache 服务器同时运行多个 Apache 服务进程时，使用端口号区分这些 Apache 服务进程。

由于笔者主机既运行了 MySQL 服务，又运行了 Apache 服务，因此笔者主机既是 MySQL 服务器，又是 Apache 服务器。

1. 端口号：服务器上运行的网络程序都是通过端口号来识别的。一台主机上的端口号可以有 65536 个之多。典型的端口号的例子是某台主机同时登录多个 QQ 账号、同时运行多个 QQ 进程，这些 QQ 进程之间使用不同的端口号进行辨识。读者也可以将“服务器”想象成一部双卡双待（甚至多卡多待）的“手机”，将“端口号”想象成“SIM 卡槽”，每个“SIM 卡槽”可以安装一张“SIM 卡”，将“SIM 卡”想象成“Apache 服务”。手机启动后，手机同时运行了多个“Apache 服务进程”，手机通过“SIM 卡槽”识别每个“Apache 服务进程”。

2. 进程与程序的关系。读者可以简单地将程序看作是硬盘上的安装程序（如 QQ 安装程序），是静态的概念。将进程看作是 QQ 程序启动后产生的一次运行活动，是动态的概念。也就是说，程序仅需占用硬盘空间即可，而进程还需要占用内存空间及 CPU 资源。

3. 如果读者正在做一些 .NET 的实验，则会开启 IIS 服务，IIS 服务默认会占用 80 端口号，此时 IIS 一旦启动，将影响 Apache 服务的启动。通过如下步骤可停止 IIS 服务，为成功启动 Apache 服务铺平道路。

(1) 通过“开始→设置→控制面板”方式打开控制面板。

(2) 通过“管理工具→服务”方式打开服务窗口，查看系统所有服务（或者直接右键单击“我的电脑”选择“服务”选项，也可以打开服务窗口）。

(3) 在服务中找到 World Wide Web Publishing Service 服务或者 IIS Admin Service 服务，单击停止服务即可停止 IIS 服务，如图 1-3 所示。

名称	描述	状态	启动类型	登录为
World Wide Web Publishing Service	提供桌面窗口管理器启动和维护服务	已启动	自动	本地系统
Desktop Window Manager Session Ma...	添加、修改和删除以 Windows Installer (*.msi)程序包提供...	已启动	手动	本地系统
Windows Installer	通过 Internet 信息服务管理器提供 Web 连接和管理	已启动	自动	本地系统
World Wide Web Publishing Service	通过缓存常用的字体数据来优化 Windows Presentation Fo...	已启动	手动	本地系统
Windows Presentation Foundation Font ...				

图 1-3 停止 IIS 服务