



工业和信息化普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# PHP编程基础与实例教程

## PHP Fundamentals & Practices

孔祥盛 主编

- 精心筛选组织内容，严格把握深度广度
- 知识实践紧密结合，理论实际融为一体
- 设计思想融入项目，全面展现实战训练



高校系列

 人民邮电出版社  
POSTS & TELECOM PRESS



工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# PHP编程基础与实例教程

PHP Fundamentals & Practices

孔祥盛 主编



高校系列

人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

PHP编程基础与实例教程 / 孔祥盛主编. -- 北京 :  
人民邮电出版社, 2011.8  
21世纪高等学校计算机规划教材  
ISBN 978-7-115-25626-3

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(2011)第131299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、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, 可以用做各类院校相关专业的教材, 同时也是面向广大 PHP 爱好者的一本 PHP 实用参考书。

### 21 世纪高等学校计算机教材 PHP 编程基础与实例教程

- 
- ◆ 主 编 孔祥盛  
责任编辑 李海涛
  - ◆ 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 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市海波印务有限公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 21 2011 年 8 月第 1 版  
字数: 549 千字 2011 年 8 月河北第 1 次印刷

---

ISBN 978-7-115-25626-3

定价: 42.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223  
反盗版热线: (010)67171154

# 前 言

---

---

---

---

---

---

---

---

PHP 是全球最普及、应用最广泛的 WEB 应用程序开发语言之一，因为其易学易用，越来越受到广大程序员的青睐和认同。目前市场上讲述 PHP 的教程还比较少，为了满足众多 PHP 爱好者的使用需求，编者根据多年从事软件开发的经验编写了本书，奉献给广大读者。

本书在内容的编排以及章节的组织上十分考究，争取让读者在短时间内掌握 PHP 开发动态网站的常用技术和方法，从而能够快速入门。本书以“坚持理论知识够用、专业知识实用、专业技能会用”为原则，在讲解具体案例的同时，融合了软件工程、软件测试、界面设计等知识，真正做到了 PHP 与项目实训的合二为一。

本书具有如下特色。

## 1. 门槛较低

读者无需太多技术基础，就能在不知不觉中非常轻松地掌握数据库设计、软件工程以及动态网站开发相关技术。

## 2. 内容丰富、严谨

作者对 PHP 内容的选取非常严谨，一环扣一环，从一个知识点过渡到另一个知识点非常顺畅和自然，而且本书内容丰富，遵循知识的学习曲线，结合具体案例编排章节的内容，并尽量做到不留死角。

## 3. 强调实训环节与 PHP 知识的结合

以讲解 PHP 基础知识为目标，以案例的实现为载体，以不同的章节完成不同的任务为理念，采用软件工程的思想实现具体案例。

## 4. 涉及面广

软件工程、软件建模、数据库及数据库设计、界面设计、软件测试等知识在本书中均有触及。

## 5. 丰富而实用的课后习题

精选 sina、百度等知名公司面试题。

本书由孔祥盛任主编，茹蓓、张同光和付强任副主编，参加编写的人员还有李林涛、潘红、石庆民和黄宪通。其中茹蓓编写第 3 章和第 4 章，张同光编写第 5 章，付强编写第 1 章和第 2 章，李林涛编写 6.1 至 6.3.3 节，石庆民编写 6.3.4 至 6.6.2 节，潘红编写第 7 章，孔祥盛编写其余章节并负责全书统稿。

由于编者水平有限，书中难免存在不妥之处，敬请广大读者批评指正。

编者

2011 年 5 月

---

---

---

---

# 目 录

第 1 章 PHP 入门	1	2.3.3 为代码添加注释	30
1.1 PHP 概况	1	习题	30
1.1.1 PHP 的优势	1	第 3 章 PHP 表达式	31
1.1.2 PHP 的应用领域	2	3.1 常量	31
1.1.3 HTML 内嵌式的脚本语言	2	3.1.1 自定义常量	31
1.2 PHP 脚本程序工作流程	3	3.1.2 常量的内存分配	32
1.2.1 WEB 浏览器	3	3.1.3 预定义常量	33
1.2.2 HTML 简介	3	3.2 变量	33
1.2.3 PHP 预处理器	3	3.2.1 变量基本概念	33
1.2.4 WEB 服务器	3	3.2.2 变量的内存分配	34
1.2.5 数据库服务器	3	3.2.3 变量赋值方式	35
1.2.6 PHP 程序的工作流程	4	3.2.4 可变变量	37
1.3 PHP 服务器构建	5	3.3 有关变量或常量状态的函数	37
1.3.1 服务器安装前准备	5	3.3.1 数据类型查看函数	37
1.3.2 服务器安装	6	3.3.2 检查常量或变量是否定义函数	38
1.3.3 PHP 服务器启动与停止	8	3.3.3 取消变量定义 unset()函数	39
1.3.4 第一个 PHP 程序	9	3.3.4 检查变量是否为“空”	40
1.3.5 PHP 服务器配置	10	3.3.5 数据类型检查函数	42
习题	14	3.4 PHP 运算符	43
第 2 章 PHP 基础	15	3.4.1 算术运算符	43
2.1 PHP 代码基本语法	15	3.4.2 递增/递减运算符	43
2.1.1 PHP 开始标记与结束标记	15	3.4.3 赋值运算符	44
2.1.2 PHP 注释	17	3.4.4 比较运算符	45
2.1.3 PHP 语句及语句块	17	3.4.5 逻辑运算符	46
2.2 PHP 程序的组成	18	3.4.6 字符串连接运算符	47
2.2.1 关于 PHP 数据	19	3.4.7 错误抑制运算符	47
2.2.2 PHP 数据类型	19	3.4.8 条件运算符	47
2.2.3 浏览器端的数据采集	23	3.4.9 类型运算符	48
2.2.4 PHP 程序的数据采集	24	3.4.10 执行运算符	48
2.2.5 PHP 数据处理	25	3.4.11 位运算符	48
2.2.6 PHP 数据的输出	26	3.4.12 运算符优先级	49
2.3 编码规范	28	3.5 数据类型的转换	49
2.3.1 书写规范	28	3.5.1 类型自动转换	50
2.3.2 命名规范	29	3.5.2 强制类型转换	52
		习题	54

<b>第 4 章 PHP 流程控制语句</b> .....58	<b>第 6 章 PHP 的数据采集</b> ..... 103
4.1 条件控制结构 .....58	6.1 浏览器端数据的提交方式 ..... 103
4.1.1 if 语句.....58	6.1.1 GET 提交方式 ..... 103
4.1.2 if...else 语句 .....58	6.1.2 POST 提交方式 ..... 104
4.1.3 switch 语句.....60	6.1.3 GET 和 POST 混合提交方式 ..... 105
4.2 循环结构 .....62	6.1.4 两种提交方式的比较..... 106
4.2.1 while 循环语句 .....62	6.2 相对路径和绝对路径 ..... 106
4.2.2 do...while 循环语句 .....63	6.2.1 绝对路径 ..... 106
4.2.3 for 循环语句 .....63	6.2.2 相对路径 ..... 107
4.2.4 循环结构应用 .....64	6.2.3 相对路径其他概念 ..... 107
4.3 程序跳转和终止语句 .....65	6.3 使用 FORM 表单实现浏览器端的 数据采集 ..... 108
4.3.1 continue 语句 .....65	6.3.1 表单标签 <form></form> ..... 108
4.3.2 break 语句 .....65	6.3.2 表单控件 ..... 108
4.3.3 终止 PHP 程序运行 .....66	6.3.3 在表单控件中使用数组 ..... 112
习题 .....66	6.3.4 表单按钮 ..... 113
<b>第 5 章 PHP 数组</b> .....68	6.3.5 FORM 表单综合应用 ..... 114
5.1 数组的基本概念 .....68	6.4 使用\$_GET 和\$_POST “采集” 表单数据 ..... 116
5.1.1 为什么引入数组 .....68	6.5 上传文件的“数据采集” ..... 117
5.1.2 数组的分类 .....69	6.5.1 与上传相关的配置 ..... 117
5.2 一维数组的创建 .....69	6.5.2 PHP 文件上传流程 ..... 118
5.2.1 直接将变量声明为数组元素 .....69	6.5.3 预定义变量\$_FILES ..... 119
5.2.2 使用 array()语言结构创建数组 .....70	6.5.4 PHP 文件上传的实现 ..... 120
5.2.3 创建数组的注意事项 .....71	6.6 WEB 服务器端其他数据采集方法 ..... 121
5.2.4 数组元素“值”的访问 .....72	6.6.1 预定义变量\$_REQUEST ..... 121
5.3 多维数组 .....74	6.6.2 预定义变量\$_SERVER ..... 122
5.3.1 二维数组的创建 .....74	习题 ..... 123
5.3.2 二维数组元素“值”的访问 .....76	<b>第 7 章 自定义函数</b> ..... 124
5.4 数组处理函数及应用 .....77	7.1 PHP 文件间的引用 ..... 124
5.4.1 快速创建数组的函数 .....77	7.1.1 include 语言结构 ..... 124
5.4.2 数组统计函数 .....79	7.1.2 require 与 include 语言结构的 比较 ..... 125
5.4.3 数组指针函数 .....81	7.1.3 include_once 和 require_once 语言 结构 ..... 126
5.4.4 数组和变量间的转换函数 .....86	7.2 函数概述 ..... 126
5.4.5 数组遍历语言结构 .....88	7.2.1 函数的概念 ..... 126
5.4.6 数组检索函数 .....91	7.2.2 函数的分类 ..... 127
5.4.7 数组排序函数 .....93	
5.4.8 数组与数据结构 .....96	
5.4.9 数组集合运算函数 .....97	
习题 .....99	



7.3 自定义函数.....127	8.6.2 MySQL 中的附加属性.....159
7.3.1 自定义函数的定义.....127	8.6.3 创建数据库表.....159
7.3.2 自定义函数的声明和调用.....129	8.6.4 显示数据库表结构.....161
7.3.3 自定义函数的参数赋值.....129	8.6.5 删除数据库表.....162
7.3.4 变量的作用域和 global 关键字.....133	8.7 表记录的更新操作.....162
7.3.5 变量的生存周期和 static 关键字.....135	8.7.1 表记录的添加.....162
7.3.6 变量函数.....137	8.7.2 表记录的修改.....164
7.4 自定义函数综合示例.....137	8.7.3 表记录的删除.....165
7.5 return 语言结构.....138	8.8 表记录的查询操作.....165
习题.....140	8.8.1 指定字段列表及列别名.....165
<b>第 8 章 MySQL 数据库.....143</b>	8.8.2 使用谓词限制记录的行数.....166
8.1 数据库概述.....143	8.8.3 使用 from 子句指定多个数据源.....166
8.1.1 数据库.....143	8.8.4 使用 where 子句过滤记录.....168
8.1.2 关系数据库管理系统 (RDBMS).....144	8.8.5 使用 order by 子句对记录排序.....170
8.1.3 结构化查询语言 (SQL).....144	8.8.6 使用聚合函数返回汇总值.....170
8.2 数据库的设计.....144	8.8.7 使用 group by 子句对记录分组 统计.....171
8.2.1 E-R 模型.....144	8.8.8 使用 having 子句提取符合条件的 分组.....171
8.2.2 主键 (Primary Key).....146	8.9 MySQL 特殊字符序列.....171
8.2.3 实体间的关系与外键 (Foreign Key).....147	习题.....172
8.2.4 约束 (Constraint).....148	<b>第 9 章 PHP 与 MySQL 数据库.....176</b>
8.3 MySQL 简介.....149	9.1 PHP 中常用的 MySQL 操作函数.....176
8.3.1 MySQL 服务器的启动.....150	9.1.1 连接 MySQL 服务器.....176
8.3.2 连接 MySQL 服务器.....150	9.1.2 设置数据库字符集.....177
8.3.3 MyISAM 和 InnoDB 存储引擎.....151	9.1.3 关闭 MySQL 服务器连接.....177
8.4 字符集.....152	9.1.4 选择当前操作的数据库.....177
8.4.1 字符集简介.....152	9.1.5 发送 SQL 语句或 MySQL 命令.....178
8.4.2 MySQL 字符集.....152	9.1.6 遍历结果集中的数据.....180
8.4.3 MySQL 中的字符集转换过程.....153	9.1.7 MySQL 服务器连接与关闭 最佳时机.....182
8.4.4 MySQL 字符集的设置.....154	9.1.8 MySQL 服务器连接与关闭函数的 制作.....183
8.4.5 SQL 脚本文件.....156	9.2 PHP 中其他 MySQL 操作函数.....183
8.5 MySQL 数据库管理.....156	9.2.1 数据库操作函数.....184
8.5.1 创建数据库.....157	9.2.2 数据库表操作函数.....184
8.5.2 选择当前操作的数据库.....157	9.2.3 选择当前操作的数据库并 发送 SQL 语句.....185
8.5.3 显示数据库结构.....157	9.2.4 表字段操作函数.....185
8.5.4 删除数据库.....157	
8.6 数据库表的管理.....158	
8.6.1 数据类型.....158	

9.2.5 其他常用函数	186	10.6.2 PHP 分页的最简单实现	222
9.3 用户注册系统的实现	188	10.6.3 带有“分页导航条”分页的实现	223
9.3.1 用户注册系统文件组织结构	188	10.6.4 分页函数的制作	224
9.3.2 用户注册界面的实现	188	10.7 新闻发布系统软件测试	226
9.3.3 数据库的实现	189	10.8 新闻发布系统系统支持习题	227
9.3.4 制作用户注册系统所需的函数	190		
9.3.5 用户注册功能的实现	190		
9.3.6 用户登录页面的实现	193		
9.3.7 用户登录功能的简单实现	193		
9.4 SQL 注入	194		
习题	195		
<b>第 10 章 新闻发布系统的开发</b>	<b>197</b>	<b>第 11 章 PHP 会话控制</b>	<b>228</b>
10.1 新闻发布系统开发流程	197	11.1 页面间的参数传递	228
10.1.1 MIS 的开发流程	197	11.1.1 重定向的概念	228
10.1.2 新闻发布系统的开发流程	197	11.1.2 使用 JavaScript 实现重定向	229
10.2 新闻发布系统的系统规划	197	11.1.3 使用 PHP 实现重定向	229
10.2.1 新闻发布系统的目标	198	11.2 会话控制	230
10.2.2 新闻发布系统的可行性分析	198	11.2.1 会话控制基本概念	230
10.2.3 新闻发布系统的项目进度表	198	11.2.2 会话控制的实现	230
10.2.4 新闻发布系统的人员分工	198	11.3 Cookie 会话技术	230
10.3 新闻发布系统的系统分析	198	11.3.1 浏览器的 Cookie 设置	230
10.3.1 新闻发布系统的功能需求	199	11.3.2 Cookie 的工作原理	230
10.3.2 新闻发布系统的技术需求	200	11.3.3 Cookie 分类	232
10.3.3 新闻发布系统中使用的模型	201	11.3.4 PHP 使用 Cookie 的步骤	232
10.3.4 新闻发布系统的 E-R 模型	201	11.3.5 创建 Cookie	234
10.3.5 新闻发布系统的数据流程图	202	11.3.6 预定义变量\$_COOKIE	235
10.3.6 数据字典 (Data Dictionary)	205	11.3.7 删除浏览器端的 Cookie	235
10.4 新闻发布系统的系统设计	206	11.3.8 新闻发布系统用户管理功能的实现 (一)	236
10.4.1 系统流程图	206	11.3.9 Cookie 数组	238
10.4.2 程序流程图	207	11.4 Session 会话技术	239
10.4.3 数据库物理设计	208	11.4.1 Session 工作原理	239
10.4.4 图形用户界面设计	209	11.4.2 Session 和 Cookie 的区别	240
10.5 新闻发布系统系统实施	209	11.4.3 PHP 使用 Session 的步骤	240
10.5.1 文件组织结构	209	11.4.4 Session 在 php.ini 文件中的配置	241
10.5.2 数据库的实施	210	11.4.5 开启 Session	242
10.5.3 新闻管理和评论管理功能的实施	211	11.4.6 Session ID、Session name 和 SID	242
10.6 分页原理及实现	221	11.4.7 预定义变量\$_SESSION	245
10.6.1 分页原理	221	11.4.8 删除和销毁 Session	245
		11.4.9 Session 的综合应用	246
		11.4.10 新闻发布系统用户管理功能的实现 (二)	248



11.4.11 新闻发布系统权限控制的实现	249	12.3.6 优化分页函数代码	294
11.4.12 Session 数组	251	习题	296
11.5 header()函数的使用	253	<b>第 13 章 新闻发布系统的</b>	
11.5.1 页面重定向	253	<b>页面美工</b>	300
11.5.2 创建 Cookie	254	13.1 JavaScript 脚本语言	300
11.5.3 服务器响应内容的控制	254	13.2 FCKeditor 在线编辑器	301
11.5.4 浏览器缓存的远程控制	258	13.2.1 FCKeditor 使用前的准备工作	301
11.5.5 常用的浏览器缓存控制函数	262	13.2.2 FCKeditor 类的成员变量和成员	302
11.5.6 完善新闻发布系统文件下载	263	方法	302
功能	263	13.2.3 FCKeditor 的高级配置	302
习题	265	13.2.4 FCKeditor 在新闻发布系统中的	304
<b>第 12 章 字符串处理</b>	267	应用	304
12.1 字符串的指定方法	267	13.2.5 FCKeditor 的文件管理	306
12.1.1 使用单引号指定字符串	267	13.2.6 FCKeditor 瘦身	308
12.1.2 使用双引号指定字符串	267	13.3 新闻发布系统页面布局的实现	308
12.1.3 使用定界符指定字符串	269	13.3.1 DIV+CSS 概述	308
12.1.4 字符串中的字符处理	269	13.3.2 界面布局图	308
12.2 字符串处理函数	270	13.3.3 使用 DIV 实现页面布局	309
12.2.1 字符串修剪函数	270	13.3.4 准备图片素材	310
12.2.2 字符串长度函数	278	13.3.5 CSS 热身	311
12.2.3 子字符串操作函数	279	13.3.6 header 层的实现	314
12.2.4 字符串比较函数	283	13.3.7 pagebody 层的实现	316
12.2.5 字符串连接和分割函数	285	13.3.8 footer 层的实现	316
12.2.6 字符串替换函数	285	13.4 新闻发布系统静态和动态页面的	
12.2.7 URL 处理函数	286	嵌入	317
12.2.8 其他常用的字符串函数	291	13.4.1 将用户管理功能嵌入到	
12.3 新闻发布系统中的字符串处理函数的		login 层	317
应用	292	13.4.2 修改 menu 层代码	318
12.3.1 删除模糊查询中关键字两边的		13.4.3 将主要功能嵌入到	
空格	292	mainfunction 层	319
12.3.2 修改文件下载功能的代码	292	13.4.4 sidebar 层和 mainbody 层的高度自	
12.3.3 修改发表评论功能的代码	293	适应功能	324
12.3.4 优化新闻列表显示功能的代码	293	13.4.5 防止图片太宽撑破 mainbody 层	326
12.3.5 模糊查询时关键字以加粗倾斜		习题	326
格式显示	294		

# 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 PHP:Hypertext Preprocessor 单词组合的首字母缩写，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 SUN 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 Oracle、MS-Access、MySQL、Microsoft SQL Server 在内的大部分数据库，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 主要用于以下 3 个应用领域。

(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 等工作。

(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。

“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。“<br/>”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“<br/>”后，将在 WEB 浏览器产生一次换行。

“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（例如“2010年3月9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 WEB 浏览器显示这些文本信息。

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。

PHP 程序文件名中的扩展名通常使用“.php”，例如 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 服务器的功能是解析 HTTP，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保存执行结果。

### 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的 Internet Explorer（简称 IE）浏览器、Mozilla 的 Firefox 浏览器等。

### 1.2.2 HTML 简介

HTML 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“<br/>”，将“<br/>”标记解析为一个换行。在 PHP 程序开发过程中，HTML 主要负责页面的互动、布局和美观。

### 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，功能是解析 HTTP。当 WEB 服务器接收到浏览器的一个 HTTP 动态请求时，WEB 服务器会调用与请求对应的程序，程序经 PHP 预处理器解释执行后，WEB 服务器向浏览器返回 HTTP 响应，该响应通常是一个 HTML 页面。浏览器接收到该 HTTP 响应后，将执行结果显示在浏览器或进行其他处理。

常见的 WEB 服务器有微软的 Internet Information Server（IIS）服务器、IBM 的 WebSphere 服务器、开源的 Apache 服务器等。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。本书将使用 Apache 服务器部署 PHP 程序。



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解释为 HTML 静态页面，然后再将解释后的静态页面返回给浏览器进行显示。

### 1.2.5 数据库服务器

数据库服务器（DataBase Server）是一套为应用程序提供数据管理服务的软件，这些服务包

括数据管理服务（例如数据的添加、删除、修改、查询）、事务管理服务、索引服务、高速缓存服务、查询优化服务、安全及多用户存取控制服务等。

常见的数据库服务器有甲骨文的 Oracle、微软的 SQL Server、IBM 的 DB2、SAP 的 Sybase 以及 SUN 公司的 MySQL 数据库服务器。由于 MySQL 具有体积小、速度快、免费等特点，许多中小型 WEB 系统选择 MySQL 作为数据库服务器。本书将选用 MySQL 讲解有关 PHP 应用程序中数据库开发方面的知识。

## 1.2.6 PHP 程序的工作流程

PHP 程序的工作流程如图 1-1 所示，具体步骤如下。

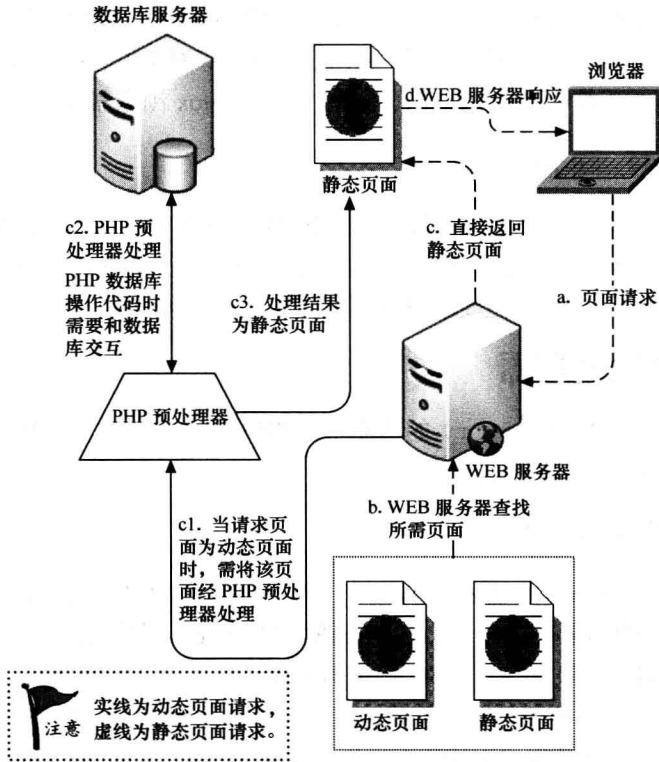


图 1-1 PHP 程序的工作流程

(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`），回车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后缀（例如 .html 或 .php）判断当前请求为静态页面请求还是动态页面请求。

当请求页面为静态页面时（例如请求页面文件名后缀为 .html 或 .htm），直接将 WEB 服务器中的静态页面返回（步骤 c），并将该页面作为响应发送给浏览器（步骤 d）。

当请求页面为动态页面时（例如请求页面文件名后缀为 .php），

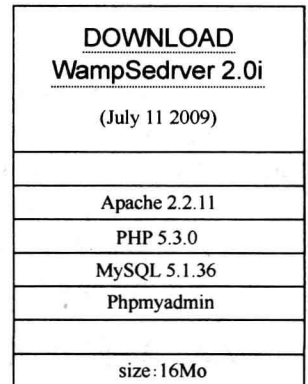


图 1-2 WampServer 下载页面

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）。

## 1.3 PHP 服务器构建

为了构建 PHP 服务器，在服务器的选择上，我们选择免费开源的 WEB 服务器 Apache 和数据库服务器 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，更快地进入 PHP 编程的殿堂。

目前两款常用的 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、布署时，通常使用 Linux 操作系统。

### 1.3.1 服务器安装前准备

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，极大方便了 PHP 服务器的安装配置和使用。WampServer 下载的官方网址是：<http://www.wampserver.com/en/download.php>，目前 WampServer 的较新版本是 WampServer2.0i。打开该网站地址后，会看到如下警告信息。

WARNING : do not try to install WampServer 2 over WAMP5. If WAMP5 is installed on your computer, save your data, uninstall it and delete the WAMP5 directory before installing WampServer 2.（警告：不要试图在 WAMP5 上安装 WampServer 2。如果 WAMP5 已经安装在计算机上，请保存程序和数据，然后卸载 WAMP5。）

还会看到“**DOWNLOAD WampServer 2.0i**”超链接（如图 1-2 所示）。图中 WampServer 2.0i 安装程序使用的 WEB 服务器为 Apache2.2.11（后面的数字为软件版本号），PHP 预处理器为 PHP5.3.0，数据库服务器为 MySQL5.1.36。



Apache 服务器在启动时，默认会占用 80 端口号；MySQL 服务器在启动时，默认会占用 3306 端口号。当这两个端口号被占用时，相应的服务器进程则会启动失败。在服务器安装启动之前，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，记录下对应进程运行时的 PID（进程唯一标识符），然后执行命令“taskill PID”，杀死该进程，释放该进程所占用的端口号，确保 PHP 服务器安装和启动成功。



以笔者主机为例，启动迅雷软件后，该软件默认会占用 80 端口号，这样有可能导致 Apache 服务器启动失败，使用下面的步骤即可停止占用 80 端口号的迅雷进程。

- (1) 选择“开始→运行”命令，打开“运行”窗口。
- (2) 在“运行”窗口中输入 cmd 命令，单击“确定”按钮。
- (3) 在命令提示符窗口中输入命令“netstat -aon”并按“Enter”键，如图 1-3 所示。

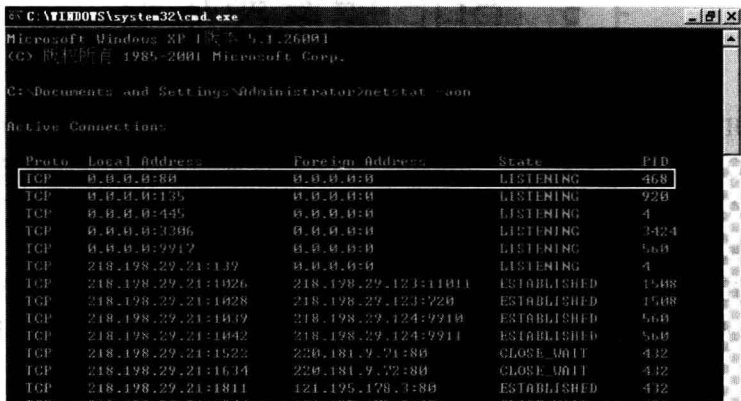


图 1-3 命令 netstat-aon 的显示效果

(4) 在 Local Address 列中查找占用 80 和 3306 端口号的程序，并记录对应的 PID。

(5) 由于当前启动的迅雷程序占用了 80 端口号，其对应的 PID 值为 468，可以在命令提示符窗口中输入命令“taskkill /F /PID 468”杀死该迅雷进程。

(6) 在命令提示符窗口中重新输入命令“netstat -aon”，即可看到迅雷进程已经被杀死，为成功启动 Apache 服务器铺平道路。

## 1.3.2 服务器安装

下载了 WampServer 安装程序，并进行了服务器安装前的准备工作后，就可以开始服务器的安装和配置了。

(1) 双击 WampServer2.0i.exe，进入 WampServer 程序安装欢迎界面，如图 1-4 所示。

(2) 单击“Next”按钮，出现许可条款界面，如图 1-5 所示。

(3) 选中“I accept the agreement”（我同意条款）单选按钮，单击“Next”按钮，出现选择安装安装路径界面，如图 1-6 所示。WAMPServer 默认的安装路径是“C:\wamp”，可以单击“Browse...”（浏览）按钮选择安装路径，这里使用默认安装路径。

(4) 单击“Next”按钮，出现创建快捷方式选项界面，如图 1-7 所示，其中第一个复选框负责在快速启动栏中创建快捷方式，第二个复选框负责在桌面上创建快捷方式。

(5) 单击“Next”按钮，出现信息确认界面，如图 1-8 所示。

(6) 信息确认无误后，单击“Install”（安装）按钮，安装接近尾声时会提示选择默认的浏览



图 1-4 欢迎界面

器，如果不确定使用哪款浏览器，单击“打开”按钮就可以了，此时选择的是 Windows 操作系统默认的 IE 浏览器，如图 1-9 所示。

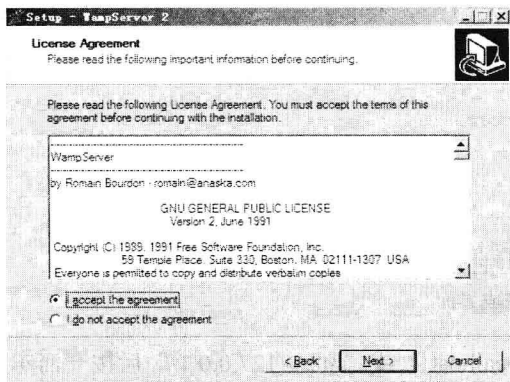


图 1-5 许可条款界面

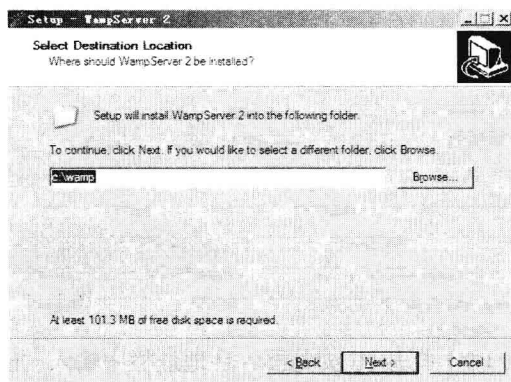


图 1-6 选择安装安装路径界面

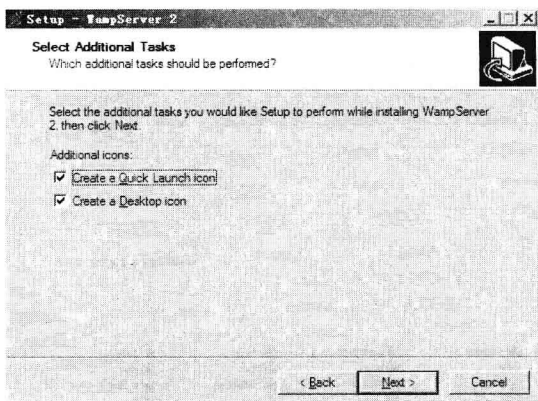


图 1-7 创建快捷方式选项界面

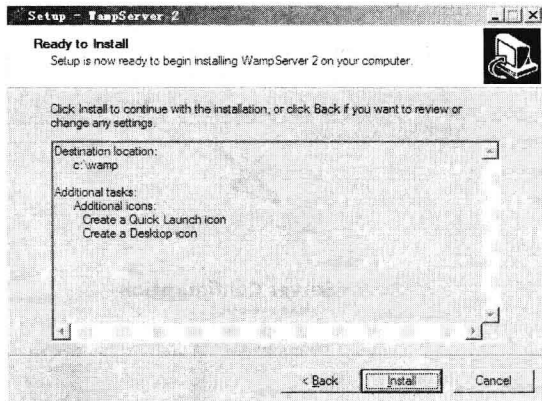


图 1-8 信息确认界面

(7) 后续操作会提示输入一些 PHP 的邮件参数信息，这里保留默认的内容就可以了，如图 1-10 所示。单击“Next”按钮将进入完成 WAMP Server 安装界面，如图 1-11 所示。



图 1-9 选择默认的浏览器

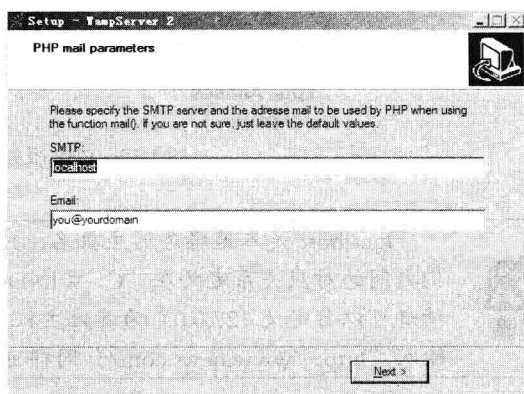


图 1-10 PHP 的邮件参数界面

(8) 当选中“Launch Wamp Server2 now”复选框时，单击“Finish”按钮后完成所有安装步骤，然后自动启动 WampServer 所有服务，并且任务栏的系统托盘中增加了 WampServer 图标。



图 1-11 安装完成界面

(9) 打开 IE 浏览器，在地址栏中输入“http://localhost/”或“http://127.0.0.1/”后按“回车”键，若出现如图 1-12 所示界面，说明 PHP 服务器安装并启动成功（图 1-12 所示界面对应的是“C:\wamp\www”目录下的 index.php 文件）。

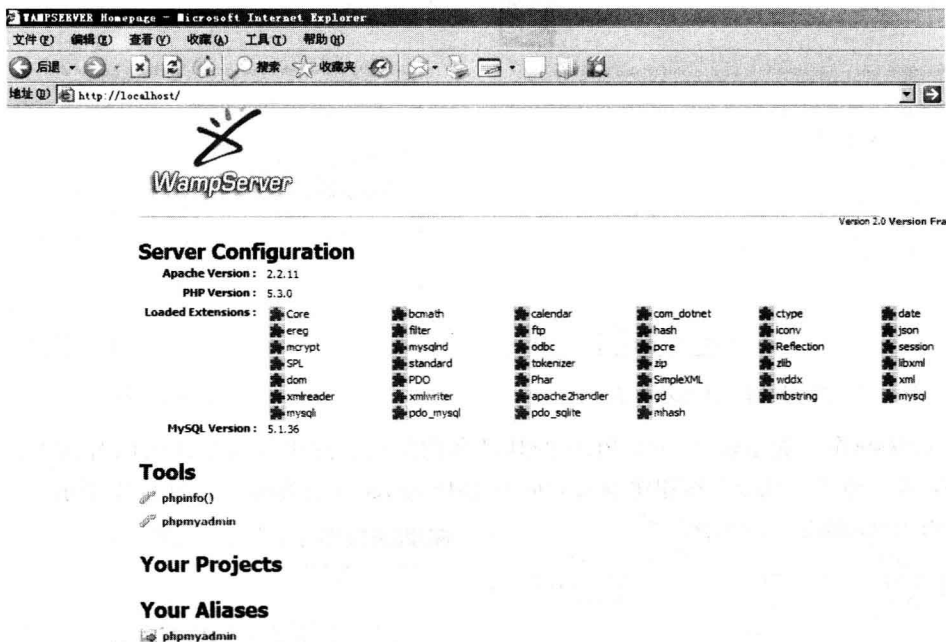


图 1-12 PHP 服务器安装且启动成功界面



注意

localhost 是本地服务器主机名，127.0.0.1 是本机地址。在 Windows 操作系统中，它们之间的对应关系定义在“C:\WINDOWS\system32\drivers\etc”目录下的 hosts 文件中。读者可以自定义 127.0.0.1 为其他主机名（例如 www.news.com），此时在浏览器地址栏中输入“http://www.news.com/”同样可以看到图 1-12 所示界面。

### 1.3.3 PHP 服务器启动与停止

PHP 服务器包括 Apache 服务器和 MySQL 服务器，读者可以选择“手动启动 PHP 服务”或