

高等学校应用型本科创新人才培养计划指定教材
高等学校计算机类专业“十三五”课改规划教材



PHP 2

程序设计及实践

青岛英谷教育科技有限公司 编著



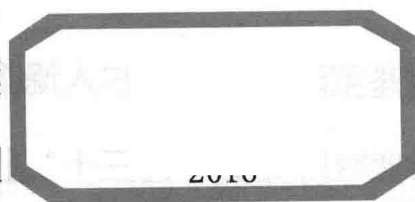
西安电子科技大学出版社
<http://www.xduph.com>

高等学校应用型本科

才

高等学校计算机类专业

才



PHP 程序设计及实践

青岛英谷教育科技有限公司 编著

西安电子科技大学出版社

内 容 简 介

PHP 简单易学且功能强大, 本书分为理论篇与实践篇, 系统地介绍了有关 PHP 程序开发的相关知识。理论篇介绍了 PHP 的基本概念、运行环境、语法规则、常用数据类型与操作、MySQL 数据库及 ThinkPHP 框架应用等相关知识; 实践篇介绍了使用 PHP 开发网站新闻发布系统的一套完整流程, 包括如何配置开发环境、搭建系统框架、编写前台页面及后台功能以及如何使用 ThinkPHP 框架开发新闻发布系统等。

本书重点突出、内容精练、实用性强。通过理论讲解与案例实践, 读者可以迅速理解并掌握最新的 PHP 开发知识与技巧, 全面提高实际动手能力。

本书适用面广, 可作为本科计算机科学与技术、软件工程、网络工程、计算机软件、计算机信息管理、电子商务等专业的程序设计课程的教材, 也可作为科研、程序设计等人员的参考书籍。

图书在版编目(CIP)数据

PHP 程序设计及实践/青岛英谷教育科技有限公司编著。

—西安: 西安电子科技大学出版社, 2016.12

高等学校计算机类专业“十三五”课改规划教材

ISBN 978-7-5606-4364-9

I. ① P… II. ① 青… III. ① PHP 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字(2016)第 299859 号

策 划 毛红兵

责任编辑 万晶晶 毛红兵

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2016 年 12 月第 1 版 2016 年 12 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 23

字 数 545 千字

印 数 1~3000 册

定 价 56.00 元

ISBN 978-7-5606-4364-9/TP

XDUP 4656001-1

如有印装问题可调换

高等学校计算机专业类 “十三五”课改规划教材编委会

主编 王 燕

编委	王成端	薛庆文	孔繁之	李 丽
	张 伟	李树金	高仲合	吴自库
	陈龙猛	张 磊	吴海峰	郭长友
	王海峰	刘 斌	禹继国	王玉锋

❖❖❖ 前 言 ❖❖❖

本科教育是我国高等教育的基础，而应用型本科教育是高等教育由精英化教育向大众化教育转变的必然产物，是社会经济发展的要求，也是今后我国高等教育规模扩张的重点。应用型创新人才培养的重点在于训练学生将所学理论知识应用于解决实际问题，这主要依靠课程的优化设计以及教学内容和方法的更新。

随着我国信息技术的迅猛发展，社会对具备信息技术能力的人才需求急剧增加，“全面贴近企业需求，无缝打造专业实用人才”是目前高校计算机专业教育的革新方向。为了适应高等教育体制改革的新形势，积极探索适应 21 世纪人才培养的教学模式，我们组织编写了高等院校计算机专业系列课改教材。

该系列教材面向高校计算机专业应用型本科人才的培养，强调产学研结合，经过充分的调研和论证，并参照多所高校一线专家的意见，具有系统性、实用性等特点，使读者在系统掌握计算机知识的同时，提高综合应用能力和解决问题的能力。

该系列教材具有如下几个特色：

1. 以培养应用型人才为目标

本系列教材以应用型软件人才为培养目标，在原有教育体制的基础上对课程进行了改革，强化“应用型”技术的学习。使读者在经过系统、完整的学习后能够达到如下目标：

- ✧ 掌握信息系统开发所需的理论和技术体系以及系统开发过程规范体系；
- ✧ 能够熟练地进行信息系统设计和编码工作并具备良好的自学能力；
- ✧ 具备一定的项目经验，包括代码调试、文档编写、软件测试等；
- ✧ 达到信息技术企业的用人标准，做到学校学习与企业的无缝对接。

2. 以新颖的教材架构来引导学习

本系列教材采用的教材架构打破了传统的以知识为标准编写教材的方法，引导读者在学习理论知识的同时，加强动手能力的训练。

教材内容的选取遵循“二八原则”，即重点内容由企业中常用的 20% 的技术组成。每章设有“本章目标”，以明确本章学习重点和难点。章节内容结合示例代码，引导读者循序渐进地理解和掌握这些知识和技能，培养学生的逻辑思维能力，掌握信息系统开发的必备知识和技巧。

另外，本系列教材借鉴了软件开发中的“低耦合，高内聚”的设计理念，组织结构上遵循软件开发中的 MVC 理念，即在保证最小教学集的前提下可以根据自身的实际情况对整个课程体系进行横向或纵向裁剪。

3. 提供全面的教辅产品来辅助教学实施

为充分体现“实境耦合”的教学模式，方便教学实施，该系列教材配备可配套使用的

项目实训教材和全套教辅产品。

✧ 实训教材：集多线于一面，以辅助教材的形式，提供适应当前课程(及先行课程)的综合项目，遵循系统开发过程，进行讲解、分析、设计、指导，注重工作过程的系统性，培养读者解决实际问题的能力，是实施“实境耦合”教学的关键环节。

✧ 立体配套：为适应教学模式和教学方法的改革，本系列教材提供完备的教辅产品，主要包括教学指导、实验指导、电子课件、习题集、实践案例及相应的网络教学资源。在教学实施方面，本系列教材提供全方位的解决方案(课程体系解决方案、实训解决方案、教师培训解决方案和就业指导解决方案等)，以适应信息系统开发教学过程的特殊性。

本书由青岛英谷教育科技股份有限公司编写。参与本书编写工作的有王万琦、宁维巍、宋国强、侯方超、何莉娟、邵作伟、王千、杨敬熹、刘江林等。本书在编写期间得到了各合作院校的专家及一线教师的大力支持与协作。在此，衷心感谢每一位老师与同事为本书出版所付出的努力。

由于编者水平有限，书中难免有不足之处，欢迎大家批评指正！读者在阅读过程中发现问题，可以通过邮箱(yinggu@121ugrow.com)发给我们，以期进一步完善。

本书编委会

2016年8月

◆◆◆ 目 录 ◆◆◆

理 论 篇

第 1 章 PHP 初步认识与环境搭建	3	3.2 变量	32
1.1 概述	4	3.2.1 变量的声明	32
1.2 PHP 新特性	4	3.2.2 变量的命名	33
1.3 PHP 的应用领域	6	3.2.3 变量的类型	34
1.4 PHP 扩展库	7	3.2.4 可变变量	39
1.5 创建 PHP 程序	9	3.2.5 变量的引用赋值	39
本章小结	9	3.3 常量	40
本章练习	10	3.3.1 设置 PHP 常量	40
第 2 章 HTTP 和 Web 服务器	11	3.3.2 预定义常量	41
2.1 资源	12	3.3.3 魔术常量	41
2.1.1 URL 语法	12	3.4 类型转换	42
2.1.2 相对 URL 和自动扩展 URL	13	3.4.1 自动类型转换	43
2.1.3 对资源的映射及访问	16	3.4.2 强制类型转换	43
2.2 事务	16	3.4.3 变量类型的测试函数	44
2.3 报文	17	3.5 运算符	45
2.3.1 方法	18	3.5.1 算术运算符	45
2.3.2 状态码	19	3.5.2 字符串运算符	46
2.3.3 首部	21	3.5.3 逻辑运算符	46
2.4 连接	24	3.5.4 比较运算符	47
2.5 Web 服务器	25	3.5.5 赋值运算符	49
2.5.1 Apache 架构	25	3.5.6 引用赋值	49
2.5.2 服务器种类	26	3.5.7 三元运算符	50
2.5.3 处理流程	27	3.5.8 错误运算符	51
本章小结	27	3.6 流程控制结构	51
本章练习	28	3.6.1 分支结构	51
第 3 章 PHP 基本语法	29	3.6.2 循环结构	55
3.1 PHP 语言标记	30	本章小结	56
3.1.1 开始和结束标记	31	本章练习	56
3.1.2 指令分隔符	31	第 4 章 字符串和数组	57
		4.1 字符串	58

4.1.1 定义方式.....	58
4.1.2 字符串实现原理.....	59
4.1.3 解析字符串.....	60
4.2 字符串处理函数.....	61
4.2.1 字符实体转换函数.....	61
4.2.2 字符串查找函数.....	64
4.2.3 字符串的子字符串操作函数.....	66
4.2.4 字符串比较函数.....	69
4.2.5 字符串通用处理函数.....	70
4.2.6 加密解密函数.....	72
4.3 数组.....	74
4.3.1 数组的类型.....	74
4.3.2 数组声明.....	74
4.3.3 设置错误报告级别.....	76
4.3.4 输出数组变量.....	78
4.3.5 数组追加及属性个数.....	78
4.3.6 数组遍历.....	79
4.3.7 二维数组.....	81
4.4 数组处理函数.....	82
4.4.1 数组创建函数.....	82
4.4.2 数组统计函数.....	83
4.4.3 数组指针函数.....	83
4.4.4 数组、变量间的转换函数.....	83
4.4.5 数组遍历语言结构.....	84
4.4.6 数组检索函数.....	84
4.4.7 其他函数.....	85
本章小结.....	85
本章练习.....	86
第5章 函数	87
5.1 函数的定义.....	88
5.2 函数的分类.....	88
5.3 函数参数传递.....	89
5.3.1 值传递.....	89
5.3.2 引用传递.....	90
5.4 变量函数.....	90
5.5 函数的作用域和生存周期.....	91
5.5.1 全局变量和局部变量.....	91
5.5.2 生存周期.....	92

5.6 文件包含.....	93
本章小结.....	93
本章练习.....	94
第6章 文件和目录	95
6.1 文件系统概述.....	96
6.2 文件的基本操作.....	96
6.2.1 打开和关闭文件.....	96
6.2.2 读取文件内容.....	97
6.2.3 写入文件.....	99
6.2.4 复制和移动文件.....	99
6.3 目录操作函数.....	101
6.3.1 打开/关闭目录.....	101
6.3.2 目录处理.....	102
6.4 上传文件.....	103
本章小结.....	106
本章练习.....	106
第7章 正则表达式	107
7.1 正则表达式简介.....	108
7.2 正则表达式语法.....	108
7.2.1 量词.....	109
7.2.2 定位符.....	109
7.2.3 限定符.....	109
7.2.4 元字符.....	110
7.2.5 模式修饰符.....	110
7.3 正则表达式引擎原理.....	111
7.3.1 占有字符和零宽度.....	111
7.3.2 正则引擎.....	111
7.4 通用字符匹配规则.....	112
7.5 正则表达式高级应用.....	112
7.5.1 零宽先行断言.....	112
7.5.2 零宽后行断言.....	113
7.5.3 分组.....	114
7.5.4 非捕获元与后向引用.....	114
7.6 关于贪婪原则和最少原则.....	115
7.7 正则表达式的函数.....	115
7.8 电子邮件验证小案例.....	117
本章小结.....	117

本章练习	118	10.3 安装 ThinkPHP	153
第 8 章 类和对象	119	10.3.1 ThinkPHP 的环境需求	153
8.1 面向对象的基本概念	120	10.3.2 ThinkPHP 的结构	153
8.2 面向对象的三大特点	120	10.3.3 入口文件的编写	154
8.2.1 封装	120	10.4 ThinkPHP 配置文件	154
8.2.2 继承	124	10.5 控制器	155
8.2.3 多态	127	10.5.1 命名规则	155
8.3 抽象类和方法(abstract)	129	10.5.2 使用规则	155
8.4 接口(interface)	129	10.5.3 使用 ThinkPHP 实现 九九乘法表	156
本章小结	130	10.6 模型	157
本章练习	130	10.6.1 命名规范	157
第 9 章 PHP 和 MySQL	131	10.6.2 连接数据库	158
9.1 PHP 操作 MySQL 数据库	132	10.6.3 实例化模型	158
9.1.1 连接 MySQL 服务器	132	10.6.4 属性访问	159
9.1.2 选择数据库文件	133	10.6.5 创建数据对象	160
9.1.3 执行数据库操作	133	10.6.6 连贯操作	161
9.1.4 从结果集中获取信息	134	10.6.7 CURD 操作	162
9.1.5 获取结果集中的记录数	136	10.7 视图	164
9.2 PDO 数据库抽象层	136	10.7.1 模板定义	165
9.2.1 PDO 构造函数	136	10.7.2 模板赋值	165
9.2.2 PDO 中的事务处理	137	10.7.3 模板输出	166
9.2.3 预处理语句	138	10.7.4 模板替换	167
9.2.4 直接执行 SQL 语句	140	10.8 ThinkPHP 的模板引擎	167
9.2.5 PDO 中获取结果集	140	10.8.1 变量输出	167
9.2.6 捕获错误	141	10.8.2 内置标签	168
9.3 使用 MySQLi	145	10.9 ThinkPHP 的单字母方法	171
9.3.1 MySQLi 面向对象	145	10.9.1 A 方法: 实例化控制器	171
9.3.2 MySQLi 面向过程	146	10.9.2 R 方法: 直接调用控制器的 操作方法	171
9.3.3 使用 MySQLi 存取数据	146	10.9.3 C 方法: 设置和获取配置参数	172
9.3.4 预准备语句	147	10.9.4 L 方法: 设置和获取语言变量	173
9.3.5 多个查询	149	10.9.5 N 方法: 计数器	174
本章小结	150	10.9.6 G 方法: 调试统计	174
本章练习	150	10.9.7 U 方法: URL 地址生成	174
第 10 章 ThinkPHP 框架	151	10.9.8 I 方法: 安全获取系统 输入变量	175
10.1 ThinkPHP 框架概述	152	10.10 ThinkPHP 的注意事项	176
10.2 ThinkPHP 框架的特点	152	10.10.1 ThinkPHP 的命名规则	176

10.10.2 ThinkPHP 页面跳转与重定向	176
本章小结	178

实 践 篇

实践 1 安装 PHP 开发环境	181
实践指导	181
实践 1.1 安装 AppServ	181
实践 1.2 安装 Zend Studio	184
实践 1.3 创建一个 PHP 项目	184
实践拓展	187
汉化 Zend Studio 软件	187
拓展练习	188

实践 2 PHP 基本语法	189
实践指导	189
实践 2.1 用户登录功能	189
实践 2.2 用户注册功能	193
实践拓展	195
三元运算符的使用	195
拓展练习	196

实践 3 字符串和数组	197
实践指导	197
实践 3.1 设计新闻前台首页	197
实践 3.2 设计新闻列表页	205
实践 3.3 设计新闻详情页	210
实践拓展	214
使用 for 循环遍历数组	214
拓展练习	215

实践 4 PHP 与 MySQL	216
实践指导	216
实践 4.1 应用 MySQL 的登录注册功能	216
实践 4.2 应用 MySQL 的新闻浏览功能	219
实践 4.3 设计搜索功能	233
实践拓展	239
MySQL 的语句执行顺序	239

本章练习	178
------	-----

拓展练习	240
实践 5 表单验证与文件处理	241
实践指导	241
实践 5.1 实现注册页校验功能	241
实践 5.2 实现评论功能	246
实践拓展	255
实现图片上传功能	255
拓展练习	259

实践 6 应用 ThinkPHP 框架开发新闻发布系统——后台设计	260
实践指导	260
实践 6.1 搭建 ThinkPHP 框架	260
实践 6.2 设计登录功能	262
实践 6.3 设计后台页面布局	267
实践 6.4 设计新闻分类管理功能	291
实践 6.5 设计新闻发布管理功能	300
实践 6.6 设计评论管理功能	310
实践 6.7 设计广告管理功能	315
实践拓展	330
使用 D 方法自动验证表单	330
拓展练习	333

实践 7 应用 ThinkPHP 框架开发新闻发布系统——前台设计	334
实践指导	334
实践 7.1 设计新闻网站浏览页面	334
实践 7.2 设计新闻网站登录注册页面	351
实践拓展	357
URL 重写	357
拓展练习	358

理论篇



第1章 PHP 初步认识与环境搭建



本章目标

- 了解 PHP 的发展史
- 掌握 PHP 的特性
- 了解 PHP 的发展趋势
- 掌握 PHP 扩展库的概念和使用
- 掌握服务器环境的搭建过程
- 掌握 PHP 的程序结构



PHP 最早发布于 1995 年, 至今已经有 21 年的历史。因其具有开发效率高、可扩展性强、开源自由等特点, 目前已成为许多中小型企业首选的开发语言。本章将介绍 PHP 语言的发展及特点, 让读者对其有一个直观的了解。

1.1 概述

PHP(Hypertext Preprocessor, 超文本预处理器)是一种开放源代码的多用途脚本语言, 可以嵌入到 HTML 中, 广泛应用于 Web 程序开发, 可以运行在 Windows、Linux 等绝大多数操作系统上。PHP 语言具有开发速度快、运行效率高、安全性好、可扩展性强、开源自由等特点, 常与开源免费的 Web 服务器 Apache 和 MySQL 数据库配合使用于 Linux 平台(简称 LAMP), 号称“Web 架构的黄金组合”, 具有最高的性价比。

PHP 由 Rasmus Lerdorf 于 1995 年首次发布, 最初名为 PHP/FI, 当时只是一套简单的 Perl 脚本, 用于跟踪访问个人主页的用户信息。1998 年 6 月正式发布的 PHP 3.0 则提供了强大的可扩展性, 并为用户提供了数据库、协议和 API 的基础结构。PHP 3.0 的可扩展性吸引了大量的开发人员加入并提交新的模块, 这是其取得巨大成功的关键。除此之外, PHP 3.0 还具备了其他的一些重要功能, 包括面向对象的支持和更强大的语法结构。

1998 年底, 鉴于旧的运行架构已不能满足 Web 程序开发的需要, Andi Gutmans 和 Zeev Suraski 计划重新编写 PHP 代码, 目标是增强运行时的性能与代码的模块化程度, 以期能对复杂程序进行高效处理。1999 年该计划成功实现预期目标, 名为“Zend Engine”(Zend 是 Zeev 和 Andi 的缩写)的引擎首次被引入 PHP。

2000 年 5 月, 基于 Zend Engine 引擎并结合了更多新功能的 PHP 4.0 发布了官方正式版本。除具有更高的性能外, PHP 4.0 还包含了一些关键性的改进, 如支持更多的 Web 服务器、增加 HTTP Sessions 支持、实现输出缓冲、采用更安全的处理用户输入的方法等。

2004 年 7 月, PHP 5 发布正式版本, 其核心是 2 代 Zend 引擎, 并引入了新的对象模型和大量新功能。

1.2 PHP 新特性

PHP 起源于自由软件, 由 C 语言改写而来, 发展极为迅速。PHP 在开发 Web 应用程序方面有如下新特性:

1. 入门快

PHP 语法简单, 并与 ASP 有部分相似, 如果读者有 C 语言或 Perl 语言基础, 则很容易上手。PHP 还有很多成熟的开发工具, 如 PHP Coder 或者 Zend Studio, 在 Linux 平台上则可以使用 Eclipse。

2. 跨平台

PHP 能在所有的主流操作系统上运行, 包括 Linux、Windows、Mac OS X、RISC OS 及 Unix 的各种变种(如 HP-UX、Solaris 和 OpenBSD)等。同时, PHP 也支持大多数的 Web 服务器, 包括 Apache、Microsoft Internet Information Server(IIS)、Personal Web Server(PWS)、



Netscape 以及 iPlant server、Oreilly Website Pro Server、Caudium、Xitami、OmniHTTPd 等。

3. 支持高度多样的数据库与协议服务

PHP 最显著的特性是支持超大范围的数据库, 涵盖类型如下: Adabas D、dBase、Empress、FilePro(只读)、Hyperwave、IBM DB2、Informix、Ingres、InterBase、FrontBase、mSQL、Direct MS-SQL、MySQL、ODBC、Ovrimos、Oracle(OCI7 和 OCI8)、PostgreSQL、SQLite、Solid、Sybase、Velocis 和 Unix dbm。

除了提供对 MySQL 与 MySQLi API 的支持外, PHP 支持的 PDO(一种 PHP 数据对象)可以帮助程序员自由地选择各种类型数据库, 而且 PHP 还支持 ODBC(Open Database Connection Standard, 开放数据库连接标准), 因此能够连接支持该标准的任何其他数据库。

PHP 支持诸多协议, 如 LDAP、IMAP、SNMP、NNTP、POP3、HTTP、COM(Windows 环境)等, 使用开放原始网络端口则可与其他任何协议协同工作。PHP 能够与所有 Web 开发语言进行 WDDX 复杂数据交换, 支持与 Java 对象的即时连接, 甚至可以用 CORBA 扩展库访问远程对象。

PHP 对正则表达式的处理功能非常强大, 从 POSIX 扩展、Perl 正则表达式到 XML 文档解析都能得到很好的实现, 这使 PHP 在文本处理上具有无可比拟的优势。PHP 4 支持使用 SAX 和 DOM 标准解析和访问 XML 文档, 也支持使用 XSLT 扩展库转换 XML 文档。PHP 5 进一步增强了 XML 文档的处理功能。它基于强大的 libxm2(C 语言的 XML 程序库)对所有 XML 扩展进行了标准化, 并添加了对 SimpleXML 与 XMLReader 的支持。

PHP 还有很多其他扩展库, 能够实现多种有趣的功能, 如 mnoGoSearch 搜索引擎函数、IRC 网关函数、多种压缩工具(gzip、bz2、zip)、日历转换器、翻译器等。

4. 良好的安全性

PHP 作为一种功能强大的开发语言, 无论以模块还是 CGI 的方式安装, 其解释器都可以在服务器上执行访问文件、运行命令及创建网络连接等操作。这或许会给服务器添加很多不安全因素, 但只要正确地安装和配置 PHP 并且编写安全的代码, 那么相对于 Perl 和 C 来说, PHP 可以在可用性和安全性之间找到一个很好的平衡点并创建出更安全的 CGI 程序。

由于 PHP 开发的程序可能用于不同的领域, 因此 PHP 内置了很多选项方便用户对其进行配置。虽然这一功能使 PHP 能够完成多样化的工作, 但是对这些选项的设定以及对服务器的配置等操作很可能产生安全问题, PHP 的选项与其语法一样, 具有很高的灵活性。如何在功能和安全性上找到一个平衡点, 这取决于 PHP 开发者对实际情况的理解。

5. 免费开源, 限制性小

PHP 拥有更多的共享性, 且不会出现被合作商限制等问题, 从而大大降低了使用成本并增强了拓展性。最经典的 PHP 开发组合当属 LAMP——Linux+Apache +MySQL + PHP。该组合上手容易, 开发速度快, 而且其中所有软件都是开源免费的, 投入很低, 因而非常适合中小型 Web 应用的开发。

6. 执行速度快, 开发、配置及部署相对简单

执行速度快: 作为一种强大的 CGI 脚本语言, PHP 的语法混合了 C、Java 和 Perl 的



新特性,且由于采用静态编译的方式,网页执行速度比 CGI、Perl 和 ASP 更快,这是它的一个突出特点。PHP 4.0 更是实现了引用计数技术,使其能够占用较少的内存消耗,更有效地使用内存。

编译速度快: PHP 是解释性的脚本语言,代码写完后即可执行,不像 C、Java、C++ 等语言需要编译后才可执行,比较节省时间。

配置简单: PHP 自身的配置文件与 Web 服务器(如 Apache、Nginx 或 Lighttpd 等)的配置都相对简单,而且修改配置之后,不需重启 Web 服务器就可立即生效。

性能优越: PHP 在 Linux/Unix 平台上运作时比在 Windows 平台上运作的性能高 45% 且能与很多免费软件组成非常省钱的开发组合,如 LAMP(Linux/Apache/Mysql/PHP)或 FAMP(FreeBSD/Apache/Mysql/PHP)。

7. 有很多成熟的框架和资源

PHP 可以使用许多开源资源,其中比较知名的开源框架如 Zend Framework、CakePHP、CodeIgniter、Symfony 等;开源论坛如 Discuz!、Phpwind 等;开源博客如 WordPress;开源网店系统如 Ecshop、ShopEx;开源的 SNS 系统如 UHome、ThinkSNS 等。

1.3 PHP 的应用领域

PHP 主要应用于以下三类领域:

1. 服务器端脚本编写

服务器端脚本编写是 PHP 最传统,也是最主要的应用领域。PHP 脚本在服务器端运行时需要具备三个必备成员:PHP 解析器(CGI 或者服务器模块)、Web 服务器和 Web 浏览器。运行时可先在 Web 服务器上安装并配置 PHP,然后使用 Web 浏览器访问 PHP 程序,即浏览服务器端的 PHP 页面。

2. 命令行脚本编写

使用命令行模式编写的 PHP 脚本不需要任何服务器或浏览器,仅使用 PHP 解析器就可以解析执行,对依赖 Cron(Unix 或 Linux 环境)或 Task Scheduler(Windows 环境)运行的脚本来说,命令行模式是最理想的选择。命令行脚本也可以处理简单的文本。

3. 桌面应用程序编写

对于拥有图形界面的桌面应用程序来说,PHP 或许并非最好的开发语言,但如果用户精通 PHP,又希望在客户端程序中使用 PHP 的一些高级特性,则可使用 PHP-GTK(PHP 的一个扩展,通常并不包含在发布的 PHP 包中)编写这些程序,该方法同样适用于编写跨平台的应用程序。

目前,全球 5000 万个互联网网站的 60%以上使用了 PHP 技术。以 CMS(内容管理系统)建站系统为例,使用率最高的三个系统分别是份额为 54.3%的第一名 Wordpress、份额为 9.2%的第二名 Joomla 和份额为 6.8%的第三名 Drupa,而这三个产品都是用 PHP 开发的。PHP 也入选了全球五大最受欢迎编程语言,并且是唯一入选的脚本语言。

在中国,80%以上的动态网站都由 PHP 开发。在 Alexa(网站排名工具)排名 TOP500



的中国网站里,有 394 家使用了 PHP 技术,所占比为 78.8%。越来越多的新公司或新项目使用 PHP,使其相关社区越来越活跃,反过来又促使更多项目或公司选择它,形成良性循环。目前 PHP 仍然是中国大部分 Web 项目开发的首选,不断有公司从其他语言(如 ASP、JAVA 等)转入 PHP。

PHP 具有运行快、开发成本低、周期短、后期维护费用低、开源产品丰富等优点,这些特性都是很多其他语言无法比拟的,也符合现阶段互联网的发展趋势。因此近几年来,PHP 日益被各类企业广泛使用,市场对专业开发人才的需求量急剧攀升,一直处于供不应求的局面。鉴于良好的就业机会和发展前景,学习 PHP 是一个不错的选择。

1.4 PHP 扩展库

扩展库可以提供 PHP 本身不具备的功能,也可为其动态地添加功能,一般位于 PHP\ext 目录下(在 PHP 4 中则位于 PHP\extensions 目录下)。在 Windows 系列操作系统中,扩展库通常记为“php_*.dll”(*代表某个具体扩展的名字);而在*nix(Unix、Linux 及其变种)系统中,则通常记为“php_*.so”或“php_*.a”。

PHP 发行包中已包含了大多数开发者最常用的扩展,它们被称为“核心”扩展库。如果用户在其中找不到能提供所需功能的扩展,则可以前往 PECL(PHP Extension Community Library, PHP 扩展社区库)中查找。PECL 相当于一个 PHP 扩展的储存室,提供了所有已知 PHP 扩展的下载途径与开发指南。

载入一个 PHP 扩展最常用的方式是在 php.ini 配置文件中声明它。需要注意的是很多扩展已经内置在 php.ini 文件里了,用户仅需要移除前面的分号来开启它们,示例如下:

```
;extension=php_extname.dll
```

```
extension=php_extname.dll
```

部分 Web 服务器会混淆这些扩展库,因为它们不一定使用和 PHP 可执行文件处于同一目录下的 php.ini 文件。而如果要搞清楚服务器具体使用的是哪个 php.ini 文件,可以在激活一个扩展后,保存 php.ini 文件并重启 Web 服务器,然后使用 phpinfo()查看。如果该扩展未在 phpinfo()中显示,则应查看日志并确定问题所在。最常见的问题是 DLL 文件位置出错(即 php.ini 中 extension_dir 的值设置错误)以及编译时的设置不匹配。如果问题是前者,则可能是下载的 DLL 文件与设置不符,可尝试重新下载与设置匹配的扩展。擅用 phpinfo()也会对纠错有很大帮助。

如果在命令行模式下使用 PHP(CLI),扩展加载出错时会在屏幕上显示信息。

1. php.ini 的常用扩展配置

因为 PHP 的扩展库需要实现对一些必要功能的扩展和支持,所以必须开启配置文件中的相关扩展,即移除这些包含在 php.ini 中的默认扩展前的“;”。下面是部分常用的需开启扩展:

- ✧ extension=php_gd2.dll: 图形处理扩展,广泛应用于实现上传头像处理、在线照片处理、验证码处理等功能。
- ✧ extension=php_mbstring.dll: 用于多字节字符串(如中文)的处理,如果没有打