



Web应用



丛书



网站开发新动力 用

PHP5

轻松开发 Web 网站

北京希望电子出版社 总策划
龚泰宁 编 著



科学出版社
www.sciencep.com



Web应用 **编程** 丛书



网站开发新动力

用

PHP5

轻松开发 Web 网站

北京希望电子出版社 总策划
龚泰宁 编 著



科学出版社
www.sciencep.com

内 容 简 介

本书重点讲述 Web 嵌入式脚本程序设计语言 PHP5。从基于 PHP 的 Web 服务器环境建立——Apache2 + PHP5 + MySQL、PHP 语法到如何使用 PHP5 进行 Web 动态脚本生成,详细讨论了 PHP 语言的语法和 PHP 脚本程序设计技术。

全书共 15 章,分别讲解了 PHP5 的语法结构与 PHP 基础编程知识、数据库应用技术、基于 PHP5 的 Web 脚本程序开发与实践等内容。本书结构严谨有序,示例丰富。可配合上机操作,学用结合,实用性强。

本书主要面对初、中级 Web 应用设计与开发人员,对初学者有较好的指导作用,同时也可作为社会培训班教材。

图书在版编目(CIP)数据

用 PHP5 轻松开发 Web 网站 / 龚泰宁编著. —北京: 科学出版社,
2006.5

(网站开发新动力)

ISBN 7-03-016860-7

I. 用... II. 龚... III. ①PHP 语言—程序设计 ②网站—设计
IV. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2006)第 008361 号

责任编辑: 但明天 / 责任校对: 肖寒
责任印刷: 媛明 / 封面设计: 梁运丽

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencecp.com>

北京媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2006 年 5 月第 一 版 开本: 787×1092 1/16
2006 年 5 月第一次印刷 印张: 27 1/4
印数: 1-3 000 册 字数: 637 26

定价: 39.00 元

前 言

PHP 是一种跨平台的服务器端嵌入式脚本语言，语法结构类似于 C、Java 和 Perl。它是一种语法结构完善的程序设计语言，并集成了自己的特色，使 Web 开发者能快速地写出可动态生成页面的脚本。

PHP 遵循 GNU 公共许可 GPL（该许可诞生出许多流行的软件，如 Linux、Emacs），可以不受限制地获得源代码，同时可以加入自己需要的特色，这也是 PHP 持续壮大的重要原因。

PHP 自从 1994 年以来历经 10 余年不断发展壮大，目前的最新版本是 PHP 5.0.4。PHP5 在对象模型、命名空间、安全、执行性能、异常管理等诸方面比以前版本都有了很大改进，增加了很多面向对象语言的重要特性，使之完全具备现代程序语言的基本特征，并保持自己在 Web 领域的强大优势。

本书面向万维网开发人员，在介绍编程原理的同时，利用大量实例总结了万维网开发经验，涉及会话过程、数据库交互、安全性和性能优化等。无论是编程的初学者，还是具有多年编程经验的程序员，都能从本书中获取有益的知识。本书以一种清晰而简练的风格介绍 PHP5 的语法和程序设计技术，并通过大量实例演示正确使用方法和习惯用法。本书附录还给出了编程规范，以及其他非常有价值的参考材料，这些内容有助于你成为出色的 PHP 程序员。

全书共分 15 章，对 PHP5 的关键特性，相对于以前版本的改进，最新语法结构以及各种技术应用都进行了深入而详实的介绍。从内容结构上可以分成五部分，分别是 PHP5 基础编程知识、数据库应用技术、PHP5 高级应用技术、PHP5 实践篇以及编程规范和参考资料等内容。全书结构严整有序，以用户所要完成的任务为基础，以提出问题、提出解决方案、分析方案、注意事项的形式进行全面系统的介绍。

另外，通过与 PHP 的先前版本进行对比，使读者对 PHP5 的特点达到较为全面、深入的理解。在学习过程中，如果适当配合上机实践，相信会达到更好的学习效果。

编者力图使本书讲解明晰，示例丰富。所介绍的每个知识点独立成篇，读者既可以根据需要选择学习，也可以自行安排学习顺序。由于编写时间较为仓促，书中难免会有疏漏和不足之处，恳请广大读者提出宝贵意见。

本书由龚泰宁和陈培组织编写，其他参加本书部分编写、录排、校对工作的人员还有：于冰、邓波、田丽韞、田军、易向东、陈发吉、吴秋丽、魏茂军、张靓、蒋红霞、冯军、张溟、郑启迪、周松建、杨宏伟、刘刚、龚志翔、李志等，并由龚超对全稿进行了严格、细致的复审。

编 者

目 录

第1章 PHP5 概述.....1	2.1.6 测试.....51
1.1 PHP5 可以做什么.....1	2.2 Linux 下安装配置 PHP5.....52
1.2 PHP 发展历史.....3	2.2.1 安装 MySQL 程序.....52
1.2.1 PHP/FI.....3	2.2.2 安装 Apache 和 PHP5.....53
1.2.2 PHP3.....3	2.2.3 测试.....54
1.2.3 PHP4.....3	2.3 详解 Apache2 和 PHP5 配置文件.....55
1.2.4 PHP5.....4	2.3.1 详解 Apache2 配置文件.....55
1.3 PHP 特点分析.....4	2.3.2 HTTP 守护进程的运行参数.....56
1.3.1 HTTP 认证.....4	2.3.3 详解 PHP5 配置文件.....61
1.3.2 Cookie.....6	2.4 小结.....66
1.3.3 XForms.....7	第3章 PHP5 语法结构.....67
1.3.4 文件上传.....8	3.1 PHP 基本语法.....67
1.3.5 使用远程文件.....12	3.1.1 分析 PHP 代码块.....67
1.3.6 连接处理.....13	3.1.2 指令分隔符的使用.....68
1.3.7 数据库永久连接.....14	3.1.3 注释符的使用.....69
1.3.8 安全模式.....15	3.2 数据类型.....69
1.3.9 命令行模式.....19	3.2.1 布尔型.....70
1.4 PHP5 的特性.....25	3.2.2 整型.....71
1.4.1 向后非兼容性变更.....26	3.2.3 浮点型.....72
1.4.2 CLI 和 CGI.....27	3.2.4 字符串.....72
1.4.3 配置文件迁移.....27	3.2.5 数组.....76
1.4.4 新函数.....28	3.2.6 对象.....80
1.4.5 新指令.....30	3.2.7 资源.....80
1.4.6 数据库.....30	3.2.8 NULL.....80
1.4.7 新对象模型.....30	3.3 常量与变量.....81
1.3.8 命名空间.....45	3.3.1 常量.....81
1.5 PHP5 和其他脚本语言.....45	3.3.2 变量.....82
1.6 小结.....46	3.4 运算符和表达式.....85
第2章 安装和配置 PHP5 环境.....47	3.4.1 运算符.....85
2.1 Winodws 下安装配置 PHP5.....47	3.4.2 表达式.....88
2.1.1 安装配置 PHP.....48	3.5 流程控制.....90
2.1.2 MySQL 安装.....49	3.5.1 替代语法.....91
2.1.3 IIS 配置.....50	3.5.2 条件语句.....91
2.1.4 安装配置 Apache.....50	3.5.3 分支语句.....92
2.1.5 Zend 优化器安装.....51	3.5.4 循环语句.....94

3.5.5 其他控制语句	98
3.6 函数	102
3.6.1 函数的参数	102
3.6.2 返回值	104
3.6.3 变量函数	104
3.7 类和对象	105
3.7.1 类	105
3.7.2 继承	107
3.7.3 构造函数	107
3.7.4 PHP5 中关于类的新特性	109
3.8 小结	118
第4章 MySQL 管理	119
4.1 SQL 基础	119
4.1.1 数据库表	119
4.1.2 创建表	120
4.1.3 删除表	121
4.1.4 数据库记录操作	121
4.1.5 select 查询操作	124
4.1.6 运算函数	125
4.2 关系数据库设计概要	129
4.2.1 关系数据库概述	129
4.2.2 关系数据库的正规化	129
4.3 MySQL 基本操作	133
4.4 数据类型和数据表	134
4.4.1 MySQL 数据类型	135
4.4.2 数据表	136
4.4.3 数据表操作	137
4.5 查询	138
4.5.1 MySQL 查询	138
4.5.2 查询实例	141
4.6 创建和使用数据库	145
4.6.1 创建并使用数据库	146
4.6.2 创建数据表	146
4.6.3 转载数据到数据表	148
4.6.4 检索信息	149
4.6.5 使用多个数据表	161
4.7 获得数据库和表的信息	163
4.8 用批处理模式使用 MySQL	164

4.9 小结	165
第5章 MySQL 管理和性能优化	166
5.1 MySQL 管理与操作	166
5.1.1 用户管理	166
5.1.2 创建用户并授权	166
5.1.3 实例：管理用户	168
5.1.4 撤权和删除用户	170
5.1.5 备份数据库	171
5.2 性能优化	175
5.2.1 编译优化	175
5.2.2 优化服务器参数	176
5.2.3 表的类型	179
5.2.4 使用命令优化数据库	180
5.3 MySQL 查询分析与优化	183
5.3.1 索引的类型	184
5.3.2 单列索引与多列索引	185
5.3.3 选择索引列	186
5.3.4 分析索引效率	187
5.3.5 索引的缺点	188
5.3.6 使用临时表进行查询优化	188
5.4 小结	189
第6章 MySQL 的安全性	190
6.1 安全性问题	190
6.1.1 内部安全性	190
6.1.2 外部安全性	192
6.2 MySQL 授权表	192
6.2.1 授权表范围列	194
6.2.2 授权表权限列	194
6.2.3 管理权限	194
6.3 控制客户访问	195
6.3.1 范围列内容	195
6.3.2 查询访问验证	197
6.3.3 范围列匹配顺序	197
6.4 避免授权表风险	198
6.5 不用 grant 设置用户	199
6.6 实例	200
6.6.1 实例：添加用户	200
6.6.2 实例：使 MySQL 更安全	202

6.7 小结.....	202	9.1 关于正则表达式.....	257
第7章 数据库编程.....	203	9.1.1 正则表达式历史.....	257
7.1 访问 MySQL 的相关函数.....	203	9.1.2 正则表达式分类.....	257
7.1.1 连接数据库服务器函数.....	203	9.2 基本语法.....	257
7.1.2 关闭数据库连接函数.....	204	9.2.1 元字符.....	258
7.1.3 选择数据库函数.....	204	9.2.2 使用元字符.....	261
7.1.4 SQL 查询函数.....	205	9.3 PHP 中的正则表达式.....	262
7.1.5 数据库记录操作函数.....	206	9.3.1 ereg()函数.....	263
7.1.6 数据库级操作函数.....	208	9.3.2 字符簇.....	263
7.1.7 数据库信息函数.....	208	9.3.3 在 PHP 中使用字符簇.....	264
7.1.8 取数据库名和表名函数.....	210	9.4 小结.....	265
7.2 创建和操纵 MySQL 数据库.....	213	第10章 类和对象.....	266
7.3 PHP5+MySQL 编程实例.....	215	10.1 对象和类的基本概念.....	266
7.3.1 实例: 查看数据表.....	215	10.2 PHP 中类和对对象的使用.....	266
7.3.2 实例: 使用表单添加数据.....	218	10.2.1 定义类.....	267
7.3.3 实例: 编辑数据库中记录.....	221	10.2.2 在 PHP 中使用类.....	268
7.3.4 实例: 删除数据库中记录.....	224	10.2.3 使用对象的变量与方法.....	268
7.4 使用 ADOdb 类库进行数据库开发.....	227	10.2.4 创建一个构造器.....	269
7.4.1 安装 ADOdb.....	228	10.2.5 类的继承.....	270
7.4.2 使用脚本连接数据库.....	229	10.2.6 使用多态.....	271
7.4.3 在数据库中进行选择操作.....	230	10.2.7 使用类的静态方法.....	271
7.4.4 插入、更新记录.....	231	10.2.8 方法工厂.....	272
7.4.5 对查询进行缓存.....	231	10.2.9 设计使用 Web 页的样式类.....	272
7.4.6 处理查询结果.....	232	10.3 高级 OOP 技术.....	278
7.4.7 限制查询结果.....	233	10.3.1 序列化.....	278
7.4.8 处理事务.....	233	10.3.2 拷贝和克隆.....	279
7.5 小结.....	234	10.4 小结.....	280
第8章 使用 Cookie 和 Session.....	235	第11章 Socket 编程.....	281
8.1 使用 Cookie 技术.....	235	11.1 Socket 基础.....	281
8.1.1 什么是 Cookie.....	235	11.2 PHP5 套接字函数.....	281
8.1.2 Cookie 如何工作.....	236	11.2.1 套接字函数.....	281
8.1.3 在 PHP 中使用 Cookie.....	237	11.2.2 PHP5 套接字函数.....	282
8.1.4 实例: 记录每个用户的信息.....	239	11.2.3 实例: 一个简单的 TCP 服务器.....	282
8.2 使用 Session 技术.....	240	11.3 其他应用.....	284
8.2.1 Session 简介.....	240	11.3.1 SMTP 基础.....	284
8.2.2 PHP 中 Session 的实现.....	241	11.3.2 使用 Socket 发送 E-mail.....	285
8.3 小结.....	256	11.3.3 聊天室实现.....	292
第9章 正则表达式.....	257		

11.3.4 实例: 一个基于 Web 的新闻组浏览器	293	13.3.1 一般策略	319
11.4 PHP 支持的协议/封装协议	294	13.3.2 安装为 CGI 程序	319
11.4.1 HTTP 和 HTTPS	294	13.3.3 安装为 Apache 模块	321
11.4.2 FTP 和 FTPS	296	13.3.4 文件系统安全	322
11.4.3 PHP 输入/输出流	297	13.3.5 数据库安全	323
11.4.4 压缩流	298	13.3.6 错误报告	327
11.5 小结	299	13.3.7 使用 register_globals	329
第 12 章 模板技术	300	13.3.8 用户提交的数据	330
12.1 使用模板的优势	300	13.3.9 隐藏 PHP	331
12.1.1 分离功能和布局	300	13.3.10 更新到最新版本	332
12.1.2 避免页面元素重复	302	13.4 用 PHP 进行数据加密	332
12.2 FastTemplate	304	13.4.1 数据加密基础	332
12.2.1 FastTemplate 模板的来历	304	13.4.2 PHP 的加密功能	332
12.2.2 使用 FastTemplate	304	13.4.3 使用 Mcrypt 加密算法扩展库	335
12.2.3 实例: 编制一个表格	306	13.4.4 使用 Mhash 扩展库	337
12.2.4 模板的执行速度	308	13.5 使用 SSI	339
12.3 PHPLIB Template	308	13.5.1 SSI 基础知识	339
12.3.1 使用 PHPLIB 模板	309	13.5.2 启动 SSI	339
12.3.2 嵌套的模板	310	13.5.3 SSI 语法格式	340
12.3.3 添加模板文本	311	13.6 小结	344
12.4 比较 PHPLIB Template 和 FastTemplate	311	第 14 章 XML 应用	345
12.4.1 PHPLIB Template 和 FastTemplate 的共同点	312	14.1 XML 概述	345
12.4.2 技术的易用性	312	14.1.1 XML 元标记语言	345
12.5 小结	313	14.1.2 XML 描述的结构和语义	346
第 13 章 程序优化和安全	314	14.2 使用 PHP 的 XML 解析器 Expat	347
13.1 编写灵活有效的 PHP 代码	314	14.2.1 Expat 简介	347
13.1.1 数据库无关性	314	14.2.2 编译 Expat	348
13.1.2 界面无关性	314	14.2.3 实例: 使用 Expat 解析 XML	348
13.1.3 可移植性	316	14.2.4 产生 XML 解析器实例的函数	352
13.1.4 面向对象/函数化	316	14.2.5 对文档进行解析	353
13.2 优化 PHP5 代码	316	14.2.6 显示统计信息	354
13.2.1 安装 Zend Optimizer	316	14.2.7 实例: 用 PHP&XML 编写简单的搜索引擎	354
13.2.2 使用 Cache 提速 PHP	317	14.3 小结	356
13.2.3 Web 内容的压缩	317	第 15 章 建设简单交互网站	357
13.3 PHP5 安全	318	15.1 计划一个站点	357
		15.1.1 用 include 和 require 进行模块化	357

15.1.2 从 HTML 到 PHP	359	A.2.9 方法中参数命名	384
15.2 添加计数器	361	A.2.10 变量命名	384
15.2.1 计数函数	361	A.2.11 引用变量和函数	385
15.2.2 改进计数器	362	A.2.12 全局变量	385
15.3 添加反馈表单	365	A.2.13 定义命名/全局常量	385
15.3.1 编写输入表单和处理数据	365	A.2.14 静态变量	385
15.3.2 配置 php.ini 文件	368	A.2.15 函数命名	385
15.4 添加站内搜索引擎	368	A.2.16 错误返回检测规则	386
15.5 添加密码验证功能	369	A.3 书写规则	386
15.5.1 基于 HTTP 验证	369	A.3.1 大括号规则	386
15.5.2 使用 PHP 对用户身份 进行验证	370	A.3.2 缩进/制表符/空格规则	386
15.5.3 根据指定验证信息核实 用户身份	370	A.3.3 括号、关键词和函数规则	387
15.5.4 另一种简单的密码验证	371	A.3.4 If..Then..Else 格式	387
15.6 添加文件上传功能	372	A.3.5 switch 格式	388
15.7 添加投票功能	374	A.3.6 continue, break 和问号	388
15.8 添加动态 banner	378	A.3.7 声明块的定位	389
15.9 小结	380	A.3.8 其他规则	389
附录 A PHP5 编码规范	381	A.3.9 布尔逻辑类型	390
A.1 介绍	381	A.3.10 避免嵌入式赋值	390
A.1.1 标准化的重要性	381	A.4 帮助与共享	391
A.1.2 优点	381	A.4.1 重用你和他人的工作	391
A.1.3 缺点	381	A.4.2 建立小型代码库	391
A.1.4 讨论	381	A.4.3 创建知识库	391
A.1.5 解释	381	A.4.4 讲一个故事	391
A.2 命名规则	382	A.4.5 归档注释	392
A.2.1 合适的命名	382	A.4.6 注释结构	392
A.2.2 类命名	382	A.5 其他	394
A.2.3 方法和函数命名	382	A.6 PHP 文件扩展名	395
A.2.4 缩写词不要全部用大写字母	383	A.7 PHP 代码标记	395
A.2.5 类命名	383	附录 B PHP5 类型和比较符表	396
A.2.6 类库命名	383	附录 C 函数别名列表	398
A.2.7 方法命名	384	附录 D 保留字列表	406
A.2.8 类属性命名	384	附录 E 资源类型列表	413
		参考文献	431

第 1 章 PHP5 概述

PHP (HyperText Preprocessor, 超文本预处理器) 是生成动态网页的工具, 而且是共享软件, 可以从 PHP 官方站点 <http://www.php.net> 免费下载。PHP 遵循 GNU 公共许可 GPL (该许可诞生出许多流行的软件, 如 Linux 和 Emacs), 可以不受限制地获得源代码, 也可以加入自己需要的特色; 公开所修改的源代码与别人分享。

它大量地借用 C, Java 和 Perl 等语言的语法, 并融合 PHP 自己的特性, 使 Web 开发者能够快速写出可动态生成页面的脚本。对于一种运行在 Web 服务器中的嵌入式脚本语言, 其基本要求包括:

(1) 解释脚本模块能够很好地与 Web 服务器结合, 使得后者能够快速解释和执行脚本程序。

(2) 语法结构完整, 具备基本程序语言功能。

(3) 支持多种数据库数据的读取和存储。

(4) 广泛地支持各种 Internet 协议。

PHP 不仅完全满足嵌入式 Web 服务器脚本语言的基本要求, 而且还提供了大量的内部函数, 方便用户实现各种功能。

当前最新的 PHP 版本是 PHP 5.0.4, PHP5 及其集成的 Zend Engine 2 大大提升了 PHP 的性能和功能, 开发者可以很轻松地把 PHP4 的代码迁移到 PHP5, 而且迁移后的大部分 PHP4 源代码不需要做任何改动就可以运用。

1.1 PHP5 可以做什么

PHP5 主要用于编写服务器端的脚本程序, 所以可以用 PHP 完成任何其他 CGI 程序能够完成的工作, 例如收集表单数据、生成动态网页以及发送/接收 Cookies。但 PHP 的功能远不局限于此, PHP 脚本主要用于如下的 3 个领域:

- 服务器端脚本。这是 PHP 最传统也是最主要的目标领域, 开展工作需要具备 3 点, PHP 解析器 (CGI 或者服务器模块)、Web 服务器和 Web 浏览器。在运行 Web 服务器时, 要确保安装并配置 PHP, 然后用 Web 浏览器来访问 PHP 程序的输出, 即浏览服务器端的 PHP 页面。
- 命令行脚本。可以编写一段 PHP 脚本, 并且不需要任何服务器或浏览器来运行它。通过这种方式, 仅仅只需要 PHP 解析器来执行。对于依赖 cron (UNIX, Linux 环境) 或者 Task Scheduler (Windows 环境) 的日常运行脚本来说这是一个理想的选择。此外, 这些脚本也可以用来处理简单的文本, 请参阅第 1.3.9 节的“命令行模式”以获取更多的信息。
- 编写客户端的 GUI 应用程序。对于基于窗口式的应用程序来说, PHP 或许不是最好的编程语言。不过, 如果你精通 PHP, 并且希望在客户端应用程序中使用 PHP 的高级特性, 那么就可以利用 PHP-GTK 来编写程序, 而且可以编写跨平台的应

用程序。PHP-GTK 是 PHP 的一个扩展，通常发布的 PHP 包中并不包含它。如果你对 PHP-GTK 感兴趣，请访问其网站以获取更多的信息。

PHP 适用于所有主流操作系统，包括 Linux，UNIX 的各种变种（如 HP-UX，Solaris 和 OpenBSD），Microsoft Windows，Mac OS X 和 RISC OS。今天，PHP 已经支持大多数的 Web 服务器，包括 Apache，Microsoft Internet Information Server (IIS)，Personal Web Server (PWS)，Netscape，iPlant server，Oreilly Website Pro Server，Caudium，Xitami 和 OmniHTTPd。对于大多数服务器来说，PHP 提供了一个模块。此外，还有一些 PHP 支持 CGI 标准，使得 PHP 能够作为 CGI 处理器来工作。

使用 PHP 可以自由地选择操作系统和 Web 服务器，还可以在开发时选择使用面向过程、面向对象或者两者混和的方式来开发程序。PHP 不局限于只输出 HTML，而且还能被用来动态输出图像、PDF 文件，甚至 Flash 动画（使用 libswf 和 Ming）。还能够简便地输出文本，例如 xHTML 以及任何其他形式的 XML 文件。PHP 能够自动生成这些文件，在服务器端开辟一块动态内容的缓存，可以直接把它们打印出来，或者将它们存储到文件系统中。

PHP 强大而显著的特性之一是，它支持很大范围的数据库。你会发现，利用 PHP 编写数据库支持的网页简单得难以置信。目前，PHP 支持如下数据库：

- Adabas D Ingres Oracle (OCI7 and OCI8)
- dBase InterBase Ovrimos
- Empress FrontBase PostgreSQL
- FilePro (read-only) mSQL Solid
- Hyperwave Direct MS-SQL Sybase
- IBM DB2 MySQL Velocis
- Informix ODBC Unix dbm

同时还有一个 DBX 扩展库，使得 PHP 可以自由的使用扩展库支持的任何数据库。另外，PHP 还支持 ODBC (Open Database Connection Standard，开放数据库连接标准)，所以它可以连接任何其他支持该世界标准的数据库。

PHP 还支持利用诸如 LDAP，IMAP，SNMP，NNTP，POP3，HTTP 和 COM (Windows 环境) 等不计其数的协议服务，可以开放原始网络端口，使得任何其他协议能够与之协同工作。PHP 支持和所有 Web 开发语言之间的 WDDX 复杂数据交换。关于相互连接，PHP 已经支持了对 Java 对象的即时连接，并且可以将它们自由地用作 PHP 对象，甚至可以用 CORBA 扩展库来访问远程的对象。

PHP 具有高效率的文本处理特性，支持从 POSIX 扩展或者 Perl 正则表达式到 XML 文档解析。为了解析和访问 XML 文档，还支持 SAX 和 DOM 标准，这样可以使用 XSLT 扩展库来转换 XML 文档。

如果将 PHP 用于电子商务领域，会发现其 Cybercash payment，CyberMUT，VeriSign Payflow Pro 以及 C CVS 函数对于在线交易程序非常有用。另外还有很多其他有趣的扩展库，例如 mnoGoSearch 搜索引擎函数、IRC 网关函数、多种压缩工具 (gzip，bz2)、日历转换，翻译等。

1.2 PHP 发展历史

到目前为止, PHP 发展大致可以分为 4 个阶段, 下面逐一做简单介绍。

1.2.1 PHP/FI

PHP 继承自一个名叫 PHP/FI 的工程。它是 1995 年由 Rasmus Lerdorf 创建, 最初只是一套简单的 Perl 脚本, 用来跟踪访问主页用户的信息, 取名为 Personal Home Page Tools。随着更多功能需求的增加, 最终形成可以让用户开发简单动态 Web 的程序。其后, Rasmus 公开了 PHP/FI 的源代码, 以便每个人都可以使用它。同时大家也可以修正它的 Bug, 进一步改善它。

PHP/FI 是一个专为个人主页或表单提供解释功能的程序, 已经包含今天 PHP 的一些基本功能。它具有类似于 Perl 的变量, 自动解释表单变量, 并且可以嵌入 HTML 中。其语法本身与 Perl 很相似, 但达到的功能还是很有限, 而且简单, 甚至有些还不协调。

1997 年 PHP/FI 2.0 (即 C 语言实现) 的第二版, 在全世界已经有几千个用户和 5 万个域名安装 (大约是 Internet 所有域名的 1%), 但当时只有几个人为该工程撰写了少量代码, 所以它仍然是一个个人工程。PHP/FI 2.0 在经历了数个测试版后, 于 1997 年 11 月发布正式版本。

1.2.2 PHP3

PHP 3.0 是接近当今 PHP 语法结构的第一个版本。Andi Gutmans 和 Zeev Suraski 在开发自己的商业程序时, 发现 PHP/FI 2.0 的功能明显不足, 于是重写了代码, 这就是 PHP 3.0 的产生。经过 Andi, Rasmus 和 Zeev 的努力, 在考虑到 PHP/FI 2.0 已存在的用户群, 他们决定联合发布 PHP 3.0 作为 PHP/FI 2.0 官方的后继版本, 而 PHP/FI 2.0 的开发几乎终止了。

PHP 3.0 最强大的功能是其可扩展性。除了给最终用户提供对数据库、协议和 API 的支持外, 它的可扩展性还吸引了大量开发人员的加入并提交新的模块。后来证实, 这是 PHP 3.0 取得巨大成功的关键。PHP 3.0 中的其他关键功能包括面向对象的支持和更强大和协调的语法结构。

由于从 PHP/FI 2.0 的名称中删除了“本语言只限于个人使用”的部分, 并使用全新的语言命名——HyperText Preprocessor, 缩写成 PHP。

到 1998 年末, PHP 的安装人数近 10 000, 有大约 100 000 个网站报告使用了 PHP。在 PHP 3.0 的发展巅峰, Internet 上 10% 的 Web 服务器都安装了它。在等待 9 个月的公开测试后, 于 1998 年 6 月正式发布 PHP 3.0。

1.2.3 PHP4

在 PHP 3.0 官方发布不久, Andi Gutmans 和 Zeev Suraski 又开始了重新编写 PHP 代码, 其设计目标是增强复杂程序运行时的性能和完善 PHP 自身代码的模块化。虽然 PHP 3.0 的功能、广泛的第三方数据库以及 API 的支持, 使得这样的程序编写成为可能, 但是 PHP 3.0 并没有高效地处理如此复杂程序的能力。

而名叫 Zend Engine (Zeev 和 Andi 的缩写) 的新引擎成功地实现了设计目标, 并在 1999

年中期首次引入 PHP。至此结合更多功能的 PHP 4.0 诞生了，它在 PHP 3.0 发布的 2 年后，于 2000 年 5 月发布了正式版本。除了更高的性能以外，PHP 4.0 还包含其他一些关键性的功能，如支持更多的 Web 服务器，对 HTTP Sessions 的支持，对输出缓存（output buffering）的支持，更安全处理用户输入的方法以及一些新的语言结构。

今天已经有 10 000 名开发人员和几百万网站报告已安装了 PHP4，占整个 Internet 域名的 20%。PHP 开发小组有很多优秀的开发人员，同时还有大量的优秀人才在进行 PHP 相关工程的开发工作，如 PEAR 和 PHP 文档的工程。

1.2.4 PHP5

2004 年 7 月 PHP 开发组发布了 PHP 5.0.0，PHP5 及其集成的 Zend Engine 2 大大提升了 PHP 的性能和功能，开发者可以轻松地把 PHP4 的代码迁移到 PHP5，大多数 PHP4 源代码不需要改动。PHP5 的关键特性包括：

- Zend Engine 2 提供一个新的对象模型和大量新特性。
- 完全重新实现对 XML 的支持。
- 新的 SimpleXML 扩展，使得开发者可以像操作 PHP 对象一样轻松地操作和控制 XML。也可以于 DOM 扩展进行交互。
- 一个名为 MySQLi 的 MySQL 扩展，满足 MySQL 4.1 和以后版本。这个新的扩展包括一个面向对象的接口，并支持 MySQL 的很多新特性，诸如 Prepared 语句。
- PHP 绑定了 SQLite。
- 改进流机制，包括访问底层套接字操作的能力。
- 其他还有很多。

1.3 PHP 特点分析

下面从几个方面说明 PHP 的典型应用和技术特点，即用 PHP 进行 HTTP 认证、Cookies、Xforms 处理、文件上传处理、使用远程文件、连接处理、数据库永久连接、安全模式以及 PHP 的命令行模式。

1.3.1 HTTP 认证

PHP 的 HTTP 认证机制仅在 PHP 以 Apache 模块方式运行时才有效，因此该功能不适用于 CGI 版本。在 Apache 模块的 PHP 脚本中，可以用 header() 函数来向客户端浏览器发送 Authentication Required 信息，使其弹出一个用户名/密码输入窗口。当用户输入用户名和密码后，包含有 URL 的 PHP 脚本将会再次与预定义变量 PHP_AUTH_USER、PHP_AUTH_PW 和 AUTH_TYPE 一起被调用，这 3 个变量分别被设定为用户名、密码和认证类型。预定义变量保存在 \$_SERVER 或者 \$HTTP_SERVER_VARS 数组中。系统仅支持“基本的”认证。请参阅 header() 函数以获取更多信息。

PHP 的版本问题：Autoglobals 全局变量，包括 \$_SERVER 等，在 PHP 4.1.0 以后版本有效，\$HTTP_SERVER_VARS 则从 PHP3 开始有效。

下面是在页面上强迫客户端认证的脚本范例。

实例：HTTP认证范例

```
<?php
    if (!isset($_SERVER['PHP_AUTH_USER'])) {
        header('WWW-Authenticate: Basic realm="My Realm"');
        header('HTTP/1.0 401 Unauthorized');
        echo 'Text to send if user hits Cancel button';
        exit;
    } else {
        echo "<p>Hello {$_SERVER['PHP_AUTH_USER']}</p>";
        echo "<p>You entered {$_SERVER['PHP_AUTH_PW']} as your password.</p>";
    }
?>
```

兼容性问题：在编写 HTTP 头信息代码时请格外小心。为了对所有的客户端保证兼容性，关键字 Basic 的第一个字母必须大写为 B，分界字符串必须用双引号（不是单引号）引用；并且在头信息行 HTTP/1.0 401 中，在 401 前必须有且仅有一个空格。

在以上例程中，仅仅打印出 PHP_AUTH_USER 和 PHP_AUTH_PW 的值，但在实际运用中，你可能需要对用户名和密码的合法性进行检查，或进行数据库的查询，或从 dbm 文件中检索。

在 PHP 4.3.0 中，为了防止有人通过编写脚本来从用传统外部机制认证的页面上获取密码，当外部认证对特定页面有效，并且安全模式被开启时，PHP_AUTH 变量将不会被设置。无论如何，REMOTE_USER 可以被用来辨认外部认证的用户，即使用 \$_SERVER['REMOTE_USER'] 变量。

设置问题：PHP 用 AuthType 选项来判断外部认证机制是否有效，但这仍然不能防止有人通过未认证的 URL 来从同一服务器上认证的 URL 上偷取密码。

Netscape Navigator 和 Internet Explorer 浏览器都会在收到 401 的服务端返回信息时清空所有的本地浏览器整个域的 Windows 认证缓存。这能够有效的注销一个用户，并迫使他们重新输入自己的用户名和密码。有些人用这种方法来使登录状态“过期”，或者作为“注销”按钮的响应行为。

实例：强迫重新输入用户名和密码的 HTTP 认证。

```
<?php
function authenticate() {
    header('WWW-Authenticate: Basic realm="Test Authentication System"');
    header('HTTP/1.0 401 Unauthorized');
    echo "You must enter a valid login ID and password to access this resource\n";
    exit;
}

if (!isset($_SERVER['PHP_AUTH_USER']) ||
    ($_POST['SeenBefore'] == 1 && $_POST['OldAuth'] ==
$_SERVER['PHP_AUTH_USER'])) {
```

```

    authenticate();
}

else {
    echo "<p>Welcome: {$_SERVER['PHP_AUTH_USER']}<br>";
    echo "Old: {$_REQUEST['OldAuth']}";
    echo "<form action='{$_SERVER['PHP_SELF']}' METHOD='POST'>\n";
    echo "<input type='hidden' name='SeenBefore' value='1'>\n";
    echo "<input type='hidden' name='OldAuth'
value='{ $_SERVER['PHP_AUTH_USER']}'>\n";
    echo "<input type='submit' value='Re Authenticate'>\n";
    echo "</form></p>\n";
}

?>

```

该行为对于 HTTP 基本认证标准来说并不是必须的,因此你不能依靠这种方法。对 Lynx 浏览器的测试表明, Lynx 在收到 401 的服务端返回信息时不会清空认证文件,因此对认证文件的检查要求没有变化,只要用户点击“后退”按钮,再点击“前进”按钮,其原有资源仍然能够被访问。不过,用户可以通过按_键来清空他们的认证信息。

同时请注意,在 PHP 4.3.3 之前,由于微软 IIS 的限制,HTTP 认证无法工作在 IIS 服务器的 CGI 模式下。为了能够使其在 PHP 4.3.3 以上版本下工作,你需要编辑 IIS 的设置“目录安全”。点击“编辑”并且只选择“匿名访问”,其他所有复选框都应该留空。

另一个限制是,在使用 IIS 的 ISAPI 模式的时候,你无法使用 PHP_AUTH_* 变量,而只能使用 HTTP_AUTHORIZATION。例如,考虑如下代码:

```

list($user, $pw) = explode(':',
base64_decode( substr( $_SERVER['HTTP_AUTHORIZATION'], 6)));

```

IIS 注意事项: 要 HTTP 认证能够在 IIS 下工作,PHP 配置选项 `cgi.rfc2616_headers` 必须设置成 0 (默认值)。

如果安全模式被激活,脚本的 UID 会被加到 WWW-Authenticate 头信息的 realm 部分。

1.3.2 Cookie

PHP 完全支持 HTTP Cookies。Cookie 是一种在远程浏览器端储存数据,以此来跟踪和识别用户。你可以使用 `setcookie()` 函数来设置 Cookie。Cookie 是 HTTP 头的一部分,因此 `setcookie()` 函数必须在其他信息被输出到浏览器前调用,这和对 `header()` 函数的限制是类似的。你可以使用输出缓冲函数来延迟脚本的输出,直到按照你的需要设置好了所有的 cookies 或者其他 HTTP 头信息。

任何从客户端发送给你的 cookies 将会像 GET 和 POST 数据一样被自动的转换为 PHP 变量,该过程受 `register_globals` 和 `variables_order` 两个变量的影响。如果希望对一个 cookie 变量设置多个值,则需在 cookie 的名称后加[]符号。

在 PHP 4.1.0 及更高版本下,自动全局变量数组 `$_COOKIE` 将总是包含所有总客户端发来的 cookies 数据。在老版本的 PHP 中,当 `track_vars` 设置变量打开时(此设置自 PHP 4.0.3 后总是打开的),系统为 cookies 设置的数组变量是 `$HTTP_COOKIE_VARS`。

关于更多细节以及有关浏览器问题的注意事项, 请参阅 `setcookie()` 函数和本书后面章节。

1.3.3 XForms

Xforms 定义传统 Web 表单的一个变种, 能够在更广泛平台和浏览器, 甚至诸如 PDF 文档之类非传统介质上使用。

Xforms 的第一个关键区别是表单传递到客户端的方式。XForms for HTML Authors 包含如何创建 Xforms 的详细描述, 下面列举一个例子。

实例: 简单的Xforms搜索表单。

```
<h:html xmlns:h="http://www.w3.org/1999/xhtml"
        xmlns="http://www.w3.org/2002/xforms">
<h:head>
<h:title>Search</h:title>
<model>
  <submission action="http://example.com/search"
              method="post" id="s"/>
</model>
</h:head>
<h:body>
<h:p>
  <input ref="q"><label>Find</label></input>
  <submit submission="s"><label>Go</label></submit>
</h:p>
</h:body>
</h:html>
```

表单中显示一个文本输入框 (名称为 `q`) 和一个提交按钮。当单击提交按钮时, 表单就会发送到动作引用的页面上。

在传统 HTML 表单中, 数据作为 `application/x-www-form-urlencoded` 进行传送。而在 Xforms 中, 数据发送格式是 XML。

如果使用 Xforms, 意味着希望数据格式是 XML, 这时 `$HTTP_RAW_POST_DATA` 中包含浏览器产生的 XML 文档、传递给 XSLT 引擎或者文档解析器。

如果对数据格式不在意, 并且只想把数据加载进传统的 `$_POST` 变量中, 可以通过把属性更改为 `urlencoded-post`, 即要求客户端浏览器以 `application/x-www-form-urlencoded` 的形式发送数据。

实例: 使用Xform填充\$_POST。

```
<h:html xmlns:h="http://www.w3.org/1999/xhtml"
        xmlns="http://www.w3.org/2002/xforms">
<h:head>
<h:title>Search</h:title>
<model>
```

```

<submission action="http://example.com/search"
             method="urlencoded-post" id="s"/>
</model>
</h:head>
<h:body>
  <h:p>
    <input ref="q"><label>Find</label></input>
    <submit submission="s"><label>Go</label></submit>
  </h:p>
</h:body>
</h:html>

```

注意：到目前为止，很多浏览器不支持Xforms。在使用这种技术之前请首先检查浏览器的版本。

1.3.4 文件上传

1. POST 方法上传

PHP 能够接受任何来自符合 RFC-1867 标准的浏览器（包括 Netscape Navigator 3, Microsoft Internet Explorer 3 或者更高版本）上传的文件。PHP 的这种特性使得我们既可以上传文本文件，也可以上传二进制文件。利用 PHP 的认证和文件操作函数，你就可以控制谁有上传的权限，以及在文件上传后进行哪些处理。

相关设置请参阅 php.ini 的 file_uploads、upload_max_filesize、upload_tmp_dir 以及 post_max_size 设置选项。

请注意 PHP 也支持 PUT 方法的文件上传，Netscape Composer 和 W3C 的 Amaya 客户端使用这种方法。请参阅 PUT 方法支持以获取更多信息。

你可以建立一个特殊的表单来支持文件上传。

实例：文件上传表单。

```

<form enctype="multipart/form-data" action="_URL_" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="30000">
Send this file: <input name="userfile" type="file">
<input type="submit" value="Send File">
</form>

```

以上范例中的 URL 应该替换成指向一个 PHP 文件的真实 URL。MAX_FILE_SIZE 隐藏域（单位为字节）必须先于文件输入域，其值为接收文件的最大尺寸。同时，要保证你的文件上传表单中要有 enctype="multipart/form-data"，否则文件上传将不能工作。

警告：MAX-FILE-SIZE 的值只是对浏览器的一个建议，实际上它可以被简单地绕过，因此不要将对浏览器的限制寄希望于该值。PHP 设置中的上传文件最大值是不会失效的，但最好在表单中加上 MAX-FILE-SIZE，因为它可以避免用户再花时间等待上传大文件之后才发现该文件太大了的麻烦。

为上传文件定义的变量会根据 PHP 的版本及设置不同而不同。自动全局变量 \$_FILES