

深入 PHP 4 编程技术

麦 奇 编著



A0944015

中 国 石 化 出 版 社

内容提要

PHP 4 是一种服务器端 HTML 嵌入式脚本描述语言。其独特的语法混合了 C、Java、Perl 以及 PHP 式的新特性，可以比 CGI 或者 Perl 更快速的执行动态网页。

本书系统地介绍了 PHP 语言的基本概念、语法规则和使用 PHP 4 语言进行程序设计的方法，并通过大量的示例进行了说明。本书体系合理、概念清晰、逻辑性强、文字流畅、通俗易懂，是一本学习 PHP 4 语言的好教材。可作为大专院校和计算机培训班的教材，也可供自学使用。

图书在版编目 (CIP) 数据

深入 PHP 4 编程技术 / 麦奇编著. -北京: 中国石化出版社, 2000.6
ISBN 7-80043-897-X

I. 深… II. 麦… III. PHP 4 语言 - 程序设计
IV. C931.4

中国版本图书馆 CIP 数据核字 (2000) 第 00584 号

中国石化出版社出版发行
北京东城区安定门外大街 58 号
邮编 100011
广东出版技校彩印厂印刷
新华书店北京发行所经销

*

787×1092 毫米 1/16 27.5 印张 插页 652 千字 印 1-1900
2000 年 7 月第 1 版 2000 年 7 月第 1 次印刷
定价: 48.00 元

前 言

当今 Web 页面不仅是由 HTML 文件构成的,而且正被许多网页制作程序所代替,PHP 语言就是其中一种,它的发展是从追踪在线访问者和在 Web 上进行 SQL 查询开始的,现在如 JavaScript 一样成为了一种热门的脚本语言,在语法结构上很象是 Perl、Java、C 等语言的混合体,所以有过这些编程基础的人都可以很快掌握这门语言。

本书从最基本的概念开始,讲述了 PHP 4 的安装与配置,PHP 的语言、语法基础知识及参考元件库的内容。通过编程示例把各部分有机地联系起来,使本书既可用作学习入门的教材,又可作为编程人员的工具书。

本书共分十章并附有一个附录。

第一章从 PHP 语言的发展,语言的特点入手,着重突出 PHP 卓越的 Web 页编写能力,并以实例作为入门者的引路标,简要介绍 PHP 4 语言的构造。

第二章是对 PHP 4 的安装配置的详尽讲解,如在 UNIX 和 MS-Windows 95/98/NT 平台上的安装实例;php4.ini 文件的配置;以及使用该软件的安全防护设定等都做了具体实用的讲解。

第三章从 PHP 4 的语言基础入手,把数据类型、常量、变量、操作符这些编程基本元素做了详细讲解,以便使有或无编程基础的读者都能从中了解、学习到 PHP 语言的一些基本常识。

第四章是 PHP 语言的基本语法结构,包括流程控制编排,表达式描述,各类语句的语法规规定等都做了讲述。

第五章是 PHP 函数与类的介绍,通过这一章,读者可以学习到函数与类的基本应用。

第六章是 PHP 所用到的函数库列举,通过每一个相应的示例,可以使读者对 PHP 函数库有一个全面的学习。

第七章是 PHP 语言的应用,PHP 4 与 MySQL 的连接,有关 MySQL 数据库的介绍及安装使用,还有两个具体的实例来加以说明 PHP 与 MySQL 的综合应用。

第八章是有关 PHP 的综合实际应用与技巧,通过十五个方面的例子描述出 PHP 在 Web 网页上的实用价值。

第九章、十章是对与 PHP 相关的 HTML 语言、SQL 语言进行的补充讲解,通过这一部分的学习可以使一个 PHP 编程者成为一个真正的编程高手,也可使初学者在此有一个质的飞跃。

最后的附录是对本书所列出的 PHP 函数作一个简单索引,使读者在学习之后不必为繁多的函数名称而重新翻阅前面的内容,而专心于编程工作之中。

本书的作者是从事多年的计算机教学与研究的专业人员,对于计算机编程有着丰富的经验,编写本书,除了想使广大的计算机爱好者学习更多的知识外,针对学校教育也作了编排,使其简明易懂,知识性强,所以作为教学用书也不失为一本优秀的教材。

由于时间仓促,能力有限,错误遗漏之处在所难免,恳请广大读者予以批评指正。

编者

2000 年 6 月

目 录

第一章 PHP 语言的概述..... 1

1.1 PHP 语言的简史.....	1
1.2 PHP 语言的特点.....	2
1.3 PHP 与其它 CGI 的比较.....	2
1.4 简单的 PHP 实例介绍.....	4
1.4.1 最简单的 PHP.....	4
1.4.2 把当前的日期添加到 HTML.....	4
1.4.3 实例总结.....	4
1.5 从 HTML 中分离 PHP 的方法.....	5
1.6 程序注解.....	6
1.7 引用文件.....	6
1.8 如何编写 PHP 程序.....	6
1.9 PHP 和 C 或 Java 的比较.....	7
1.10 PHP 和 ColdFusion、Active Server Page 或 Java Server Page 的比较.....	7
1.11 PHP 的花费.....	7

第二章 PHP 4 的安装和配置..... 9

2.1 在 UNIX 下从源码安装 PHP 4.....	9
2.1.1 安装的指令序列 (Apache 模块版) 10	
2.1.2 对 php4.ini 文件的配置的方法.....	11
2.1.3 配置 PHP 4 的运行模式.....	11
2.1.4 数据库支持的配置选项.....	12
2.1.5 其他配置选项.....	14
2.2 在 Windows 95/98/NT 上安装 PHP 4.....	17
2.2.1 公共的安装步骤.....	17
2.2.2 Windows 95/98/NT 与 PWS/IIS 3.....	18
2.2.3 Windows NT 与 IIS 4.....	19
2.2.4 Windows 9X/NT 与 Apache 1.3.x.....	19
2.2.5 Omni HTTPd 2.0b1 for Windows.....	20
2.2.6 PHP 4 模块.....	20
2.3 php4.ini 文件的配置.....	20
2.3.1 一般配置指导.....	21
2.3.2 邮件配置指导.....	23
2.3.3 安全模式配置指导.....	23
2.3.4 调试器配置指导.....	23
2.3.5 扩展装载配置指导.....	24
2.3.6 MySQL 配置指导.....	24
2.3.7 mSQL 配置指导.....	24
2.3.8 Postgres 配置指导.....	24
2.3.9 Sybase 配置指导.....	24
2.3.10 Sybase-CT 配置指导.....	25

2.3.11 统一的 ODBC 配置指导.....	25
2.3.12 BC 数学函数配置指导.....	25
2.3.13 浏览器兼容配置指导.....	25
2.3.14 Informix 配置指导.....	25
2.3.15 session 有关的设置.....	26
2.4 安全问题.....	27
2.4.1 CGI binary.....	27
2.4.2 Apache 模块.....	29

第三章 PHP 的数据..... 30

3.1 数据类型.....	30
3.1.1 数字.....	30
3.1.2 文本.....	31
3.2 常量.....	32
3.3 变量.....	34
3.3.1 标量.....	35
3.3.2 数组变量.....	35
3.3.3 多维数组.....	37
3.3.4 变量替换.....	38
3.3.5 动态变量名.....	38
3.3.6 PHP 的变量类型.....	39
3.3.7 变量的使用.....	40
3.4 操作符.....	43
3.4.1 操作符的优先级.....	43
3.4.2 三重操作符.....	45
3.4.3 算术操作符.....	46
3.4.4 条件操作符.....	46
3.4.5 一元操作符.....	47
3.4.6 位操作符.....	48
3.4.7 字符串连接操作符.....	49
3.4.8 逻辑操作符.....	50
3.4.9 赋值操作符.....	52

第四章 PHP 的语句..... 54

4.1 PHP 的流程控制.....	54
4.2 表达式.....	54
4.2.1 简单表达式.....	54
4.2.2 有副作用的简单表达式.....	55
4.2.3 复杂表达式.....	55
4.3 语句.....	55
4.3.1 语句类型.....	56
4.3.2 非执行语句.....	56
4.3.3 执行语句.....	56

目 录

4.3.4 赋值语句	56	6.2.10 URL 处理函数库	151
4.3.5 判断语句	56	6.2.11 Vmailmgr 邮件处理函数库	152
4.3.6 循环语句	59	6.2.12 Session 函数库	153
4.3.7 跳转语句	63	6.2.13 FTP 文件传输函数库	153
第五章 PHP 函数与类	67	6.2.14 HTTP 相关函数库	154
5.1 函数	67	6.3 服务器函数类型	154
5.1.1 函数返回值	67	6.3.1 Apache 服务器专用函数库	154
5.1.2 向函数传递参数	69	6.3.2 Hyperwave 服务器函数库	155
5.1.3 给函数赋予缺省值	70	6.4 时间、日期和历法函数类型	157
5.1.4 控制变量的作用域	70	6.4.1 历法函数库	157
5.1.5 嵌套函数调用	71	6.4.2 日期与时间函数库	159
5.1.6 递归函数	72	6.5 文件处理函数类型	161
5.2 类	75	6.5.1 文件系统函数库	161
第六章 PHP 的函数库	77	6.5.2 FDF 函数库	175
6.1 数据库函数类型	77	6.5.3 PDF 格式文件函数库	175
6.1.1 Adabas D 数据库连结函数库	77	6.5.4 WDDX 函数库	178
6.1.2 DBA 函数库	77	6.5.5 压缩文件函数库	178
6.1.3 dBase 格式数据表函数库	78	6.5.6 XML 剖析函数库	181
6.1.4 dbm 类数据库函数库	80	6.6 图形处理函数库	183
6.1.5 FilePro 数据库函数库	83	6.7 PHP 4 选项及相关资讯函数库	194
6.1.6 Informix 数据库函数库	85	6.8 其他函数类型	196
6.1.7 InterBase 数据库函数库	86	6.8.1 阵列处理函数库	196
6.1.8 mSQL 数据库函数库	86	6.8.2 拼字检查函数库	202
6.1.9 SQL Server 数据库函数库	94	6.8.3 BC 高精度函数库	203
6.1.10 MySQL 数据库函数库	95	6.8.4 杂项函数库	205
6.1.11 Sybase 数据库函数库	104	6.8.5 信号与共享记忆体函数库	206
6.1.12 ODBC 数据库连结函数库	108	6.8.6 数学运算函数库	207
6.1.13 Oracle 8 数据库函数库	115	6.9 编码及字符串的函数类型	213
6.1.14 Oracle 数据库函数库	115	6.9.1 mcrypt 编码函数库	213
6.1.15 PostgreSQL 数据库函数库	120	6.9.2 mhash 杂凑函数库	214
6.1.16 Solid 数据库连结函数库	127	6.9.3 Perl 相容语法函数库	215
6.2 网络函数类型	127	6.9.4 常规表示法函数库	215
6.2.1 目录管理函数库	127	6.9.5 字串处理函数库	217
6.2.2 动态连结函数库	130	6.9.6 变量处理函数库	226
6.2.3 程序执行功能函数库	130	6.9.7 GNU 记录函数库	229
6.2.4 IMAP 电子邮件系统函数库	131	第七章 与数据库的连接	230
6.2.5 LDAP 目录协定函数库	139	7.1 MySQL 数据库的简介	230
6.2.6 电子邮件函数库	147	7.1.1 MySQL 的定义	230
6.2.7 网络函数库	148	7.1.2 MySQL 的历史	230
6.2.8 NIS 函数库	150	7.1.3 MySQL 的特点	231
6.2.9 SNMP 网管函数库	150	7.1.4 MySQL 的支持的操作系统	232
		7.1.5 MySQL 的获得	233

7.2 MySQL 的安装.....235	(netscape only) 319
7.2.1 MySQL 的安装布局.....235	9.1.3 背景色彩和文字色彩..... 319
7.2.2 安装 MySQL 二进制代码分发.....236	9.1.4 页面空白 (Margin) 320
7.2.3 安装 MySQL 源代码分发.....240	9.1.5 链接 (Link) 320
7.2.4 安装后期的设置和测试.....243	9.1.6 开一个新的 (浏览器) 窗口 (Target Window) 321
7.3 PHP 4 与 MySQL 的交互使用.....246	9.1.7 标尺线..... 321
7.3.1 开端.....249	9.2 字体 (FONT) 标记 (TAGS) 322
7.3.2 创建连接.....249	9.2.1 标题字体 (Header) 322
7.3.3 获取 HTML 表单信息.....250	9.2.2 字体大小..... 322
7.3.4 使用 HTML 表单信息.....251	9.2.3 物理字体 (Physical Style) 322
7.3.5 common.inc 文件.....252	9.2.4 逻辑字体 (Logical Style) 323
7.3.6 总结.....252	9.2.5 指定“字体大小”的标记和“指定字体”的标记的组合使用..... 323
7.4 基于 Web 的 MySQL 数据库管理程序.....253	9.2.6 字体颜色..... 323
7.5 PHP 4 与 MySQL 的使用两例.....271	9.2.7 客户端字体 (Font Face) 323
7.5.1 客户留言本程序.....271	9.2.8 字符实体 (Entities) 326
7.5.2 邮件列表程序.....277	9.3 文字布局 (TEXT STYLE) 标记 (TAGS) 327
第八章 综合应用与技巧..... 284	9.3.1 行的控制..... 327
8.1 网页计数器.....284	9.3.2 文字的对齐 (Alignment) 328
8.2 使用者认证.....287	9.3.3 文字的分区 (Division) 显示 328
8.3 聊天室.....292	9.3.4 列表..... 328
8.4 留言板.....295	9.3.5 定制列表元素..... 329
8.5 意见信箱.....303	9.3.6 预格式化文本 (Preformatted Text) 330
8.6 广告轮转.....305	9.3.7 空白 (Spacer) 331
8.7 使用 PHP 建立 GIF309	9.3.8 多列文本..... 331
8.8 文件上载支持.....309	9.3.9 其他 331
8.9 使用字符串函数输出整数化的 PHP 版本号.....310	9.4 图像 (IMAGE) 标记 (TAGS) 332
8.10 使用 PHP 产生一个随机密码.....310	9.4.1 链入图像的基本语法..... 332
8.11 一个可以在线编辑的文件的程序.....310	9.4.2 图像和文字的对齐..... 332
8.12 统计访问者来访次数的程序.....311	9.4.3 图像在页面中的对齐/布局 (Floating Image) 332
8.13 页面重定向的程序.....312	9.4.4 边框..... 333
8.14 动态新闻发布程序.....312	9.4.5 客户端图像映射图 (Client Side Image Map) 333
8.15 网页访问统计程序.....314	9.5 表单 (FORM) 标记 (TAGS) 334
8.15.1 MPCOUNT.TEXT 文件.....314	9.5.1 基本语法..... 334
8.15.2 MPCOUNTADMIN.PHP 文件.....315	9.5.2 复选框 (Checkbox) 和单选框 (RadioButton) 335
8.15.3 MPCOUNT.PHP 文件.....316	9.5.3 图像坐标..... 336
第九章 HTML 语言..... 319	9.5.4 隐藏表单的元素..... 336
9.1 页面 (PAGE) 标记 (TAGS)319	9.5.5 列表框 (Selectable Menu) 336
9.1.1 HTML 文件结构 (Document Structures)319	
9.1.2 语言字符集 (Charsets) 的信息	

目 录

9.6 表格 (TABLE) 标记 (TAGS)	338	10.14 以 M 开头的 SQL 函数	394
9.6.1 表格的基本语法	338	10.15 以 N 开头的 SQL 函数	401
9.6.2 跨多行、多列的表元 (Table Span)	338	10.16 以 O 开头的 SQL 函数	403
9.6.3 表格尺寸设置	338	10.17 以 P 开头的 SQL 函数	404
9.6.4 表格内文字的对齐/布局	339	10.18 以 Q 开头的 SQL 函数	405
9.6.5 表格在页面中的对齐/布局 (Floating Table)	340	10.19 以 R 开头的 SQL 函数	406
9.6.6 表格的标题	340	10.20 以 S 开头的 SQL 函数	408
9.7 多窗口页面 (Frames)	340	10.21 以 T 开头的 SQL 函数	413
9.7.1 基本语法	340	10.22 以 U 开头的 SQL 函数	415
9.7.2 纵向排列多个窗口	341	10.23 以 V 开头的 SQL 函数	416
9.7.3 横向排列多个窗口	341	10.24 以 W 开头的 SQL 函数	417
9.7.4 纵横排列多个窗口	341	10.25 以 Y 开头的 SQL 函数	417
9.7.5 各窗口间相互操作 (Frame Target)	342		
9.7.6 特殊的四类操作	342	附录 PHP 函数简述	418
9.7.7 Frame 的外观 (Appearance)	342	A	418
9.7.8 边框色彩 <frameset bordercolor=#>	343	B	418
9.7.9 浮动窗口 (Floating Frame)	344	C	418
9.8 会移动的文字 (Marquee) (IE ONLY)	345	D	419
9.8.1 基本语法	345	E	419
9.8.2 文字移动属性的设置	345	F	420
9.8.3 外观 (Layout) 设置	346	G	420
9.9 多媒体页面 (Alternative Inline Elements) (IE ONLY)	346	H	421
9.9.1 嵌入多媒体文本 (EMBED)	346	I	422
9.9.2 背景音乐	347	J	424
9.9.3 控制条 	347	K	424
		L	424
		M	425
第十章 SQL 命令、类型和函数	348	N	427
10.1 SQL 命令	348	O	427
10.2 以 A 开头的 SQL 函数	352	P	428
10.3 以 B 开头的 SQL 函数	355	Q	429
10.4 以 C 开头的 SQL 函数	356	R	429
10.5 以 D 开头的 SQL 函数	361	S	430
10.6 以 E 开头的 SQL 函数	368	T	431
10.7 以 F 开头的 SQL 函数	371	U	431
10.8 以 G 开头的 SQL 函数	386	V	431
10.9 以 H 开头的 SQL 函数	387	W	431
10.10 以 I 开头的 SQL 函数	387	X	431
10.11 以 J 开头的 SQL 函数	390	Y	432
10.12 以 K 开头的 SQL 函数	391		
10.13 以 L 开头的 SQL 函数	391		

第一章 PHP 语言的概述

1.1 PHP 语言的简史

PHP 是一种服务器端 HTML-嵌入式脚本描述语言。它是一个巢状的缩写名称,即“PHP: Hypertext Preprocessor”。PHP 是一种 HTML 内嵌式的语言(类似 IIS 上的 ASP),而 PHP 独特的语法混合了 C、Java、Perl 以及 PHP 式的新语法,它可以比 CGI 或者 Perl 更快速的执行动态网页,如图 1-1 所示的是 PHP 的作用示意图。

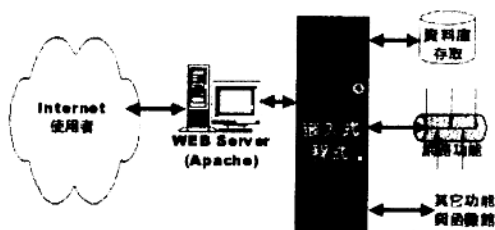


图 1-1

创始人 Rasmus Lerdorf 在 1994 年秋季开始构思 PHP。早期的非发行版本主要用以追踪谁在他的主页上,浏览他的在线简历。他在 1995 年年初推出的第一个交付用户使用的版本,由一个非常单纯的只能理解很少数特殊宏的分析引擎和一些用在主页后端通用的工具(如留言簿、计数器以及其他一些东西)组成。当时 PHP 只被认为是个人主页开发工具。

在 1995 年年中作者重写这个分析器,并被命名为 PHP/FI 第二版。FI 来源于他写的另外一个 html 表单集成数据的软件包。他把个人主页工具,表单集成工具合并在一起,并加入了 mSQL 数据库的支持。此后 PHP/FI 便以一种令人惊奇的速度传播开来,人们开始大量使用它编写程序。

据估计,在 1996 年末,整个世界至少有 15,000 个网站在用 PHP/FI。到 1997 年年中,这个数字已经超过 50,000 了。而在此时 PHP 的发展也发生了变化。由 Rasmus 自己偏爱的和几个人开发的项目变成一个更有组织的团体成就。Zeev Suraski 和 Andi Gutmans 重写了解析器。通过这次全面的重写,大量 PHP/FI 的功能被移植到 PHP 中,并且成为了 PHP 的基本雏形。

到 1998 年年中时,已经有大量的商业化产品,例如 C2 的 StrongHold web server 和 RedHat Linux 分别捆绑了 PHP/FI 解析器或 PHP 解析器,而据 NetCraft 估计的保守数据显示,PHP 已经被世界上 150,000 个站点所采用。通过进一步分析,我们知道其使用数量远比使用 Netscape 公司旗舰产品(企业级 Web 服务器)的站点多。

PHP 的第四代 Zend 核心引擎已经进入测试阶段。整个脚本程序的核心大幅更动,让程序的执行速度,满足更快的要求。在最佳化之后的效率,已较传统 CGI 或者 ASP 等程

序有更好的表现。而且还有更强的新功能、更丰富的函数库。无论您接受不接受, PHP 都将在 Web CGI 的领域上, 掀起颠覆性的革命。对于一位专业的 Web Master 而言, 它将是必修课程之一。

1.2 PHP 语言的特点

现在 PHP 可能是 Internet 上最热的脚本语言之一, 它是一种新型的 CGI 程序编写语言, 速度快, 并可同时运行于 Windows、Unix、Linux 等多种平台上。强大的功能和易用性使得它从诞生以来, 受到了无数 Web 编程者的青睐。很多用 CGI 或者其他方法实现起来很困难的任务, PHP 可以只用几行简单的代码就搞定。如同 ASP 一样, PHP 代码也是嵌入在 HTML 中的, 这给编程带来了更多的方便。

PHP 在很大程度上可以说是 Perl、Java 和 C 等诸多概念的混合体。它的语法结构基本上是借鉴于 C, 这使得 PHP 学起来并不难, 即使是新手, 也可以很容易地登堂入室。它内置了文件上传、密码认证、Cookies 操作、邮件收发、动态 GIF 生成等功能。

而它最为强劲的特性就是其数据库功能, 把一个数据库发布到网上从没变得如此简单。它支持现今大部分流行的数据库服务器, 包括 Oracle、Sybase、mSQL、MySQL、Informix、Solid dBase、ODBC、Unix dbm、PostgreSQL、Adabas D、FilePro Velocis 等等, 这样给 Web 编程者带来了很大的选择余地。

同时 PHP 不仅可以应用在 Unix 环境, 还可以应用于 Windows 环境, 除了支持 APACHE 服务器外, 还支持 PERSONAL WEB 等服务器。我们将要在后面详细地讨论 PHP 数据库接口, 并且, 将要特别地把 MYSQL 拿出来, 讨论它和 PHP 的接口问题。

PHP 一个很重要的特点是: 它是一个面向目标 (goal-oriented) 的语言, 实现任务的时候简明而且快捷。

PHP 是所谓的服务器端 (server-side) 脚本语言, 同 ASP 一样, 但和那些客户端脚本像 VBScript、JavaScript 有很大的区别。也就是说, 在 PHP 脚本能够工作之前, 它的解释程序必须已经安装在服务器上了。并且要确认配置没有错误。PHP 是一种服务器内置式的 Script 语言, 它的出现使得在 Unix 上快速地开发动态 Web 成为现实。

PHP 的解释程序可以到 <http://www.php.net/> 下载。但 PHP 已经是完全重新写过了的, 所以和 PHP 2.0 以及 PHP/F1 并不是很相同。所以一定要弄清楚你安装的到底是那个版本的 PHP。在安装和调试好 PHP 后, 保留一份说明文档是很必要的。这个文档列出了所有的命令和语法结构, 这在学习 PHP 的过程中是必不可少的。

1.3 PHP 与其他 CGI 的比较

无可置疑的, 写 CGI 的方式有很多种, 而 PHP 只是其中的一种选择罢了。对资深的 Webmaster 而言, CGI 的写作界面应是随着需求而更动。

毕竟, 在一个对系统反映速度要求极严格的系统而言, 恐怕只有 NSAPI 介面写的 CGI 程序才能符合要求了。在其他的场合, 相信使用 PHP 来作为 CGI 的介面是游刃有余, 而且是最适合的。

PHP 4 与各种 CGI 的比较如表 1-1 所示。

表 1-1 PHP 4 与各种 CGI 的比较

程序介面	PHP	ASP	CGI	NSAPI	ISAPI
作业系统	均可	Win32	均可	均可	Win32
Web 服务器	数种	IIS	均可	Netscape Server	IIS
执行效率	快	快	慢	极快	极快
稳定性	佳	中等	最高	差	差
开发时间	短	短	中等	长	长
修改时间	短	短	中等	长	长
程序语言	PHP	VB	不限	C/C++	C/Delphi
网页结合	佳	佳	差	差	差
学习门槛	低	低	高	极高	高
函式支援	多	少	不定	中等	少
系统安全	佳	极差	最佳	佳	尚可
使用站台	超多	多	多	极少	少
改版速度	快	慢	无	慢	慢

其中的 PHP 可用在数种 Web 服务器上；传统 CGI 就不限是哪种作业系统或 Web 服务器平台；NSAPI 一定要在 Netscape 的服务器（如 Netscape Enterprise Server 或 FastTrack Server）上才可以执行，但可支援多种作业系统（UNIX 或 Win32）；ASP 及 ISAPI 只在 IIS 上有完整的功能。

在稳定性上，由于 NSAPI 或 ISAPI 是动态连结的方式，因此在执行若出现问题，会使得 Web 服务器一起瘫痪。而 ASP 在吾人实际应用经验上，隔阵子就会使系统不稳定，需要重开作业系统。PHP 在许多的站台使用上，不但长期使用都没有问题，而且程序的稳定性也不错。当然最稳的还是传统 CGI 程序，因为它是由作业系统负责控制，不会因 CGI 程序的错误导致 Web 服务器的不稳定。

在开发及维护时间上，PHP 及 ASP 都有不错的表现。而 NSAPI 及 ISAPI 则需要长时间的开发过程，在稳定上线后，这二种介面反倒是效率最佳的方法。传统的 CGI 程序则要视开发工具语言而定了，用 Perl 或是 Shell Script 不需要编译的过程，直接就可以执行，若用 Delphi 或 VC/BCB 甚至用组合语言等都要经过编译才能执行，至于用 VB 来写传统 CGI，唉……

要比较和网页结合的能力，PHP 和 ASP 是并驾齐驱的，其他方式就不能内嵌 HTML 语法了。而这也是影响开发时间的因素之一。

就系统安全性而言，ASP 是最差的，在没有经过微软的 IIS Service Pack 处理过，使用 %\$DATA 就可以看到 ASP 的原始码，这真是叫人不敢领教。当然，传统 CGI 的程序，由于是由作业系统直接管理，要破解的难度最高，骇客必须由作业系统下手，而不能由 Web 服务器下手。PHP 在许多商业及非商业使用时，也没有听过有什么安全的问题。

在新增功能及改版方面，传统的 CGI 由于不受任何语言限制，没有这方面的问题。

PHP 是最有活力的，数天至数周就有一个新版本出现，每次的新版，就代表更多的功能及修正更多的错误。

其他的 ASP、NSAPI、ISAPI 就视它的 Web 服务器改版速度了，ASP 要等到 IIS 5.0 出现时才会有 ASP 3.0，也就是要等到 Windows 2000 正式上市。

总言之，在 Web 的后端 CGI 程序，就像鱼与熊掌一般，没有高效能又开发方便的选择。不过相信 PHP 是处于开发容易、效能亦不错的平衡点上。

1.4 简单的 PHP 实例介绍

1.4.1 最简单的 PHP

在 HTML 文档中加入以下的文字：

```
<P align="center"><?echo("这是 PHP 的实例显示\n");</p>
```

并且把文件名换为.PHP，在一个支持 PHP 的 Apache 的服务器上，直接在浏览器的地址栏中键入这个 URL，就可以在浏览器中得到结果。

1.4.2 把当前的日期添加到 HTML

PHP 语句能够很容易地插入到 HTML 中，这为实现动态设计提供了一个很方便的途径。比如，如果想要把当前的日期添加到 HTML 文件中，可以这样做：

```
<HTML>
<HEAD>
<TITLE>Our first PHP3.0 script</TITLE>
</HEAD>
<BODY>
<CENTER> 我的第一个 PHP3.0 脚本 </CENTER>

<?
/* 上面的 "<?" 标明从这儿开始就是 PHP 脚本了 */
$today = date("Y-m-d");
PRINT "<CENTER> 今天是 :$today.</CENTER>";

# 后面的 ">" 表示脚本结束
?>
</BODY>
</HTML>
```

把上述代码保存为文件名的后缀是.PHP3 的文件，假设今天的日期是 4 月 20 号，2000 年，那么在浏览器的地址栏中键入这个 URL，则可以看到就会有如下显示：

我的第一个 PHP 3.0 脚本
今天是：2000-04-20

1.4.3 实例总结

看了这两段脚本，可以得到以下总结：

- 1) 所有的 PHP 语句都必须用 “`<?>`” 符号，而不是 “`<“`”、`<”`” 或 `<“”`”。

2) 在脚本中可以加注释。可以用“/*”和“*/”来注释一段话,或者用在行的前面加上“#”来注释掉这一行。

3) 如果要在屏幕上输出结果,就要用到 PRINT 语句,输出的内容应该用双引号“”括起来。

4) 每条 PHP 语句都以分号“;”来结束。

5) 任何在 PRINT 中的 HTML 语句都会被客户浏览器来解释输出。

6) 包含有 PHP 语句的文件,只有用后缀 *.PHP 命名后(比如: myphpfile.PHP)才能够被 PHP 的解释器所识别并进行相应的处理。

7) date() 函数的使用方法如下:

语法: : string date(string format, int timestamp);

这个函数可以有两个参数, timestamp 参数是可选的,如果没有 timestamp 参数,则使用当前时间。然后函数根据给出的格式(由 format 参数给出)返回相应格式化了的时间字符串。在上面的例子中, Y-m-d 分别标了明年,月,日,都用数字来表示。

1.5 从 HTML 中分离 PHP 的方法

PHP 的语法主要借鉴于 C。也部分参考了 Java 和 Perl。从 HTML 中分离 PHP 程序有四种方法:

1) 用“?”,如: <? echo("this is the simplest, an SGML processing instruction\n"); ?>

2) 用“? php”,如: <?php echo("if you want to serve XML documents, do like this\n"); ?>

3) 用以下语句:

```
<script language="php">
```

```
echo("some editors (like FrontPage) don't like processing instructions");
```

```
</script>
```

4) 用<% echo ("这是类似 ASP 嵌入语法的 PHP 范例"); %>。

其中第一种及第二种是最常用的二个方法,在小于符号加上问号后,可以加也可以不加 php 三个字,之后就是 PHP 的程序码。在程序码结束后,加入问号大于二符号就可以了。

第三种方法对熟悉 Netscape 服务器产品的 Webmaster 人员而言,有相当的亲切感,它是类似 JavaScript 的写作方式。而对于从 Windows NT 平台的 ASP 投向 PHP 的使用者来说。

第四种方法似曾相似,只要用 PHP 3.0 版本以后的服务器都可以用小于百分比的符号开始,以百分比大于结束 PHP 的区段,但想用第四种方法的使用者别忘了在 php.ini 加入 asp_tags 或是在编译 PHP 时加入 --enable-asp-tags 的选项,才能使第四种方法有效。建议少用第四种方法,当 PHP 与 ASP 原始码混在一起时就麻烦了。

其实,在写作 PHP 的程序最好的方法,就是先处理好纯 HTML 格式的 Homepage 文件之后,再将需要变数或其他处理的地方改成 PHP 程序。这种方法,可以让您在开发上达到事半功倍的效果。

1.6 程序注解

在 PHP 的程序中, 加入注解的方法很灵活。可以使用 C 语言、C++ 语言或者是 UNIX 的 Shell 语言的注解方式, 而且也可以混合使用。这可以让每个写 PHP 网页程序的 Webmaster 或 Programmer 发展出属于自己的写作风格。

```
<?php
echo "这是第一种例子。\\n"; // 本例是 C++ 语法的注解
/* 本例采用多行的
   注解方式      */
echo "这是第二种例子。\\n";
echo "这是第三种例子。\\n"; # 本例使用 UNIX Shell 语法注解
```

```
?>
```

不过在使用多行注解时请注意, 不能让注解陷入巢状回圈当中, 否则会引起错误。

```
<?php
/*
echo "这是错误的示范。\\n"; /* 巢状注解会引起问题 */
*/
?>
```

1.7 引用文件

PHP 最吸引人的特色之一大概就是它的引用文件了。用这个方法可以将常用的功能写成一个函数, 放在文件之中, 然后引用之后就可以呼叫这个函数了。

引用文件的方法有二种: require 及 include。二种方式提供不同的使用弹性。

1) require 的使用方法如 “require (“MyRequireFile.php”);”。这个函数通常放在 PHP 程序的最前面, PHP 程序在执行前, 就会先读入 require 所指定引入的文件, 使它变成 PHP 程序网页的一部分。常用的函数, 亦可以这个方法将它引入网页中。

2) include 使用方法如 “include (“MyIncludeFile.php”);”。这个函数一般是放在流程控制的处理区段中。PHP 程序网页在读到 include 的文件时, 才将它读进来。这种方式, 可以把程序执行时的流程简单化。

1.8 如何编写 PHP 程序

若只会用 FrontPage、Dreamweaver 的所见即所得编辑模式来写网页, 而完全不懂 HTML 的语法, 恐怕要先下点功夫了解 HTML 语法, 才能顺利地写 PHP 程序了。若早就极度了解 HTML 幕后排版语言的语法, 那么应该可以马上开始 PHP 的程序写作了。在 PHP 这种后端服务器的程序语言, 下苦功去学习 HTML 是写作的必备条件。

对熟悉 Windows 系列作业平台的使用者, 有套软体倒是在开发 PHP 程序上是蛮方便的, 就是 PHP Editor, 若对这软体有兴趣, 不妨到 <http://www.soysal.com/PHPEd> 下载最新的版本。

同时还需要下载 PHP 程序 Win32 的版本。安装妥 PHP Win32 版后, 在 PHP Editor 中设定好 PHP Win32 的路径, 就可以让您轻松地开发 PHP 程序了。

至于在写作 PHP 程序的工作平台上, 有没有 Web 服务器倒是不重要, 因为在 PHP

Editor 的环境中写好的程序，按下执行的键后，由 PHP Editor 直接将所写的 PHP 程序送给 PHP Win32，并将执行结果处理后，呈现在使用者的面前。

若对 Linux/UNIX 很偏执，或许可以尝试使用 Bluefish：蓝鱼 PHP 写作工具，只要装好 Linux 及 X-Window 之后就可以使用了。

要使用中文的话，需要在 X-Window 加入中文的字型及输入法，要方便的话，使用 RedHat 加上 CLE 中文支援就好了。

假使您已是老 Linux/UNIX 了，也不喜欢使用这些专门的开发工具。相信您对 vi 或是 Emacs 一定相当熟悉，那么就在装妥 Web 服务器和 PHP 程序后，直接用 vi 或 Emacs 写 PHP 程序就可以了，并且可以直接看到执行的真正结果，和资料库或是其他的伺服程序连结，也不会有什么问题。这种「所写即所得」的方式，岂不痛快!!

1.9 PHP 和 C 或 Java 的比较

PHP程序和C程序有略微相似的地方，这是由于PHP来源于C、Perl和Java。一些对C语言来讲比较麻烦的方面（如处理字符串的技术），由于采用了和Perl相近的技术，在PHP中就不再是麻烦了。PHP中也不包括Java中的一些面向对象特性。

PHP能很好地处理某些低级的任务，如果代码需要运行的特别快，可以很方便的使用C为PHP程序编写模块。

1.10 PHP 和 ColdFusion、Active Server Page 或 Java Server Page 的比较

如果从流行的商议方式，总是“可以回答是，也可以回答不是”。

回答是，是由于它的费用（免费）以及可以通过邮件列表、IRC而在世界范围内有广泛的支持网络。同时PHP还在不断地发展着，如果想要一个PHP现在没有的特性时，请等待几个月，就有人很快为PHP编程而加上此种特性。

回答不是，是由于商业软件有比 PHP 更多的特性。

例如，Allaire's ColdFusion 有一个已证实可靠的搜索引擎组件。

微软的 Active Server Page 技术提供了与 Outlook、电子邮件和工作组程序相结合的能力。

Java Server Page 让使用者可以使用互联网上的大量开放源码模块。

1.11 PHP 的花费

PHP需要花费多少钱？这应该是一个网站的经营者在选择使用的过程中所考虑的问题。

值得高兴的事，PHP是免费的。

所有的源码、文档都可以免费的复制、编译、打印和分发。用户的任何一个用PHP编写的程序都属于用户自己，可以按照自己的意愿进行处理，而不需要付任何版税。就PHP本身而言，用户可以无限制地发布自己编写的程序。

然而，有着很充分的理由，PHP并不是彻底的“公共领域”代码。如果PHP源码是完全的公共领域代码，很可能就有人对PHP的源码做很小的修改，然后编译并出售，换一句话说，从它的创作者中脱离出来。

但从另一方面说，如果不公布源码，很难使想使用它的用户感到放心。

GNU 通用公共许可是一种发布免费软件的方法，它可以避免其他人占有你的劳动成果。在符合 GNU 许可的条件下，源码可以自由地发布，并且任何一个人都可以使用，但是任何来源于此代码的程序必须以同样种类的许可方式发布。换句话说，如果用户从任何 GNU 许可下的源码中得到代码而编写自己的程序，用户必须给任何想得到源码的人公布自己的源码。

第二章 PHP 4 的安装和配置

2.1 在 UNIX 下从源码安装 PHP 4

本节将讲述 PHP 4 在 UNIX 下的整个安装配置过程。首先应具备以下知识和软件：

- 1) 基本的 UNIX 技能（会操作“make”程序和一种 C 编译器）。
- 2) ANSI C 编译器。
- 3) 一个 Web Server。

也许读者以前从没有编译过Linux应用程序，在这里介绍一些基本概念，以便在出现问题时有可能诊断出问题所在。

1. tar

tar, 即tape archiver, 它可以把几个文件组合成一个文件, 并可以选择是否进行压缩。这个命令过去通常用来进行备份, 以便使数据存储在磁带中。当tar文件被压缩时, 它们有一个.gz的后缀; 当tar文件没有压缩时, 它们有一个.tar的后缀。

2. gcc

gcc 是GNU的C编译器。它的工作是把人可以看懂的源代码文件编译成机器可以读懂的目标文件。C源文件通常有.c的后缀名, 目标文件通常有.o的后缀名。如果编译工作不能正常进行, 就是碰到了一个编译期的错误, 或者说: 语法错误。在大多数情况下, 不彻底的编译通常是编译器找不到一个或几个包含文件而产生的。包含文件都有一个.h的扩展名, 通常用来定义不同的系统信息, 以及将多个不同的.c文件所共有的信息收集在一起。

3. make

make是一个常用的工具程序, 是用来帮助进行编译的。它的工作是用来只编译那些还没有编译过的源文件。编译.c文件将生成一个.o文件, 如果.c文件比.o文件更新, 也就是说到上次编译之前, 源文件被编辑过, 那么make将会重新编译.c文件。make指令一般是寻找一个Makefile文件, 在这个文件中包含有一个或多个能执行的目标, 例如, make clean会告诉make执行清除目标。

4. ld

ld是GNU的链接程序。它的工作是把所有的目标文件和库链接起来, 创建一个单一的可执行文件。幸运的是, 基本上不用手工运行这个程序, 因为Makefile将会考虑到所有的编译细节。

5. ldconfig

ldconfig会为在多个库目录(在/etc/ld.so.conf中指定)中寻找共享库。共享库常被多个的应用程序使用, 它们的文件名中的某个地方有.so, 例如, libqt.so.1.42是一个共享库。在编译完毕之后, 有可能需要在/etc/ld.so.conf文件中增加一个目录, 并且运行ldconfig -v命令。

6. ./configure

configure将会在计算机中寻找一些关键信息, 例如, 安装的是哪一种C编译器, 包含文件在哪里等等。然后, configure将会按照所用的计算机配置重新修改Makefile文件。应

该使用 `./configure` 来在当前目录下运行程序，以避免偶然运行 `$PATH` 环境变量中的目录下的其他程序。

7. 符号连接

符号连接允许使用者可以为一个已经现有的文件指定另一个不同的名字。

例如，使用者有可能想把一个名为 `libqt.so.1.42` 的文件称为 `libqt.so.1`。

符号连接基本上可以使用户把一个文件复制到另一个不同的目录下面，但事实上并没有真正复制此文件。第二个文件实例只是简单地指向第一个文件实例。

使用符号连接至少可以有两点好处，第一点是符号连接可以节省硬盘空间，少到只有 16 字节；第二点有一点微妙，让我们先看一下一个名为 `libqt.so.1` 且指向 `libqt.so.1.42` 的符号连接。

如果需要升级到 `libqt.so.1.88`，该怎么办呢？用户只要简单地改变符号连接，使 `libqt.so.1` 指向 `libqt.so.1.88`，而非 `libqt.so.1.42` 即可。这意味着涉及到 `libqt.so.1` 的程序将自动地使用这个库的新版本。按照惯例，符号连接常用于完整的发布版本号。换句话说，`libqt.so.1.88` 和 `libqt.so.2.32` 应该有两个不同的动态连接（分别为 `libqt.so.1` 和 `libqt.so.2`）。

2.1.1 安装的指令序列（Apache 模块版）

下面是如何为 Apache 服务器以模块的方式加入 PHP 4 的支持。

1) 首先输入：

```
gunzip apache_1.3.x.tar.gz
tar xvf apache_1.3.x.tar
gunzip php4-3.0.x.tar.gz
tar xvf php4-3.0.x.tar
cd apache_1.3.x
```

2) 配置编译 Apache 服务器的选项，如下示范，但在实际配置中可根据需求进行调整：

```
./configure --prefix=/www
cd ../php4-3.0.x
```

3) 下一步则是配置 PHP 4 编译的选项：

```
./configure --with-mysql --with-apache=../apache_1.3.x --enable-track-vars
```

4) 接着编译 PHP 4。

```
make
```

5) 安装 PHP 4。

```
make install
```

6) 在编译 Apache 服务器编译选项中加入 PHP 4 模块。

```
cd ../apache_1.3.x
./configure --prefix=/www --activate-module=src/modules/php4/libphp4.a
make
make install
```

你可以简单地把 `httpd` 二进制拷贝到现有的二进制上来代替这一步，并且确信你先关闭你的服务器。

7) 然后输入。

```
cd ../php4-3.0.x
cp php4.ini-dist /usr/local/lib/php4.ini
```

这时你可以在文件 `/usr/local/lib/php4.ini` 设置 PHP 4 选项，但你也可以把这个文件放