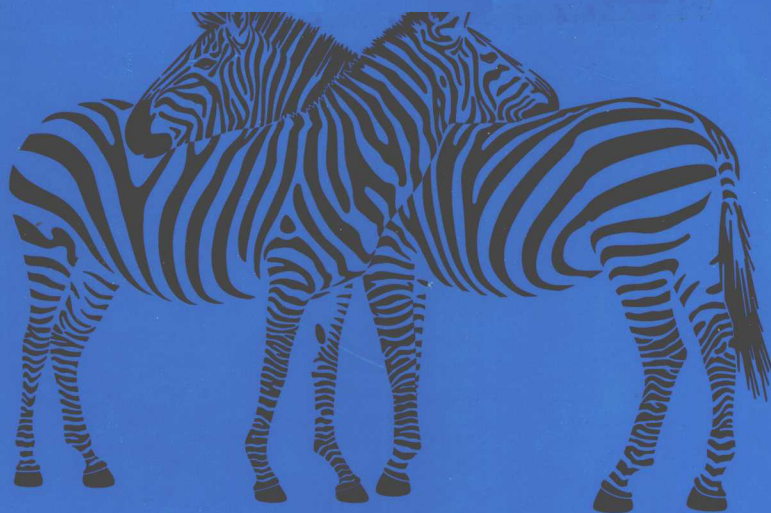


PHP5

应用实例详解：使用 **Zend Framework & Smarty**

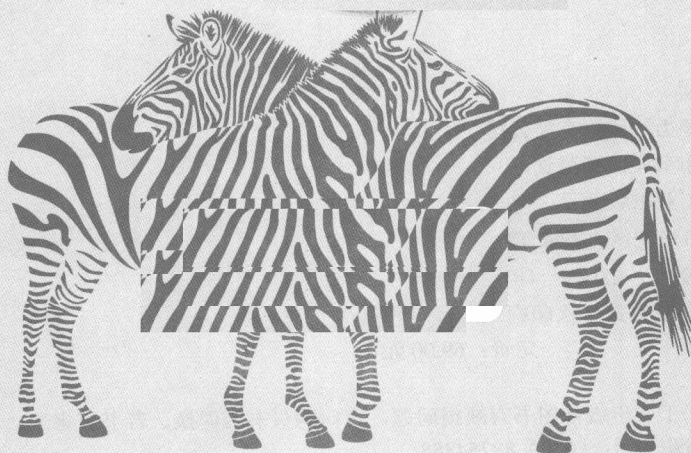
构筑真正的MVC模式应用



PHP5

应用实例详解：使用 **Zend Framework & Smarty** 构筑真正的MVC模式应用

于杰刚 朱蕾 编著



電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

PHP5 现在的性能可以说已经超过上一个版本 PHP4 了,它强化了面向对象编程的概念,同时有了 Zend Framework & Smarty 这个强大的组合,使用 PHP 也可以构筑 MVC 模式的应用系统了。本书突破以往的写作手法,直接从 33 个可以立即导入到自己的应用程序中非常实用的应用实例入手,从浅到深,将每个实例细致地剖解,让读者真正地从实践当中学习如何用 PHP 构筑网络应用系统,而且本书还花了大量的篇幅介绍如何使用 Zend Framework & Smarty 构筑真正的 MVC 模式应用系统——书签共享系统。

本书讲述简单、直接且有效,更可贵的是非常注重实用性,所以它适合所有 PHPer 当做案头必备学习参考用书。无论您是刚刚入门的新兵,还是有一定编程基础的中级读者,一定会从这本书中收获良多。

图书在版编目(CIP)数据

PHP5 应用实例详解:使用 Zend Framework&Smarty 构筑真正的 MVC 模式应用 / 王志刚, 朱蕾编著. —北京:电子工业出版社, 2010.9
ISBN 978-7-121-11433-5

I. ①P… II. ①王… ②朱… III. ①PHP 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 141146 号

责任编辑:孙学瑛

印 刷:北京智力达印刷有限公司

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×980 1/16 印张:35.25 字数:756 千字

印 次:2010 年 9 月第 1 次印刷

印 数:4000 册 定价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

笔者从业十几年以来，大部分的时间都是使用 Java 来进行应用程序的开发的。大家都知道 Java 有优良的跨平台性（Windows、UNIX 环境皆宜），因其为面向对象语言，所以代码具有良好的可维护性，且可重复使用。自从出现了 Struts 等框架（Framework）技术，引入了 MVC 模式的编程概念后，在进一步提高系统的可维护性的同时，也提供了开发的效率。因此很多应用系统采用 Java 来开发。

但是，对于那些中小系统来说，维持运行 Java 应用程序（包括服务器租用或专线租用，各种应用软件）的费用，相对来说太昂贵。尤其与 PHP 应用程序的费用相比，要高出不少。可以找很多便宜的支持 PHP 的租用服务器，但很难找到便宜的可运行 Java 应用的租用服务器。而且使用 Java 进行开发时，要比想象的费时费力。因此，对于中小应用系统来说，我个人觉得选择 PHP 比 Java 更明智。

自 2004 年 7 月发布 PHP5 至今，已经过了 5 年多了。PHP 的运行环境已经与过去有很大的不同了。当初一些人可能对 PHP5 的性能有些微词，随着 PHP5 升级到 PHP5.1、PHP5.2，性能上有了巨大的改善。PHP5 现在的性能可以说已经远远超过上一个版本 PHP4 了。对于个人用户，或一些中小应用系统来说，很多服务器空间提供商提供对 PHP5、MySQL、PostgreSQL 等的支持，可以利用这些资源快速的构筑自己的网络应用系统。

PHP5 中强化了面向对象编程的概念，同时有了 Zend Framework & Smarty 这个强大的组合，使用 PHP 也可以构筑 MVC 模式的应用系统了。由于 PHP 的易学性，使用 PHP 构筑 MVC 模式的应用系统比 Java 更有优势，特别是对于中小企业而言。尽管很多人对 PHP 还存在些安全疑虑，但我个人认为 PHP 是比 Java 更有前途的语言，特别适合我们国家的国情，可以帮助中小企业使用它推进自己的信息化。这正是笔者当初着手编写这本书的主要原因。

本书除了提供许多可以立即导入到自己的应用程序中使用的应用实例外，花了大量的篇幅（第4章的一部分，第5章）介绍如何使用 Zend Framework & Smarty 构筑真正的 MVC 模式应用系统——书签共享系统。整个论述从基础知识开始，包括系统分析、共通功能的提炼，到最后的系统合成，循序渐进。相信只要理解了本书实例以及其中介绍的内容，应该可以使用 Zend Framework & Smarty 来构筑自己的 MVC 模式的应用程序了。

本书的相关代码以及相应的开源软件可以从 www.broadview.com.cn 下载得到。

朱蕾参加了本书第1章，第2章，第3章的编写。另外，在此特别感谢我的岳父，岳母在本书编写过程中的大力支持。

编著者

2010年5月

目 录

第 1 章 导言

1.1 PHP 技术的由来.....1	1.3.2 应用程序构架 12
1.1.1 服务器端脚本的发展 轨迹——服务器端技术 以前的技术1	1.3.3 Linux 环境中的基本 环境配置..... 14
1.1.2 服务器端脚本的发展 轨迹——CGI 的时代.....3	1.3.4 Windows 环境中的基本 环境配置..... 23
1.1.3 服务器端脚本轨迹—— 服务器端脚本5	1.3.5 Linux/Windows 共同的 基本环境设置..... 39
1.1.4 各种服务器技术的比较 ...6	1.4 PHP/MySQL 的基础知识 40
1.2 PHP 的发展.....8	1.4.1 PHP 基础知识..... 40
1.3 PHP 基本运行环境的设定.....10	1.4.2 面向对象语法 45
1.3.1 Web 服务器.....11	1.4.3 PEAR 的基础知识 48
	1.4.4 MySQL 的基本知识 49

第 2 章 PHP 的基本应用

2.1 统计在线人数53	2.6 使用加密函数 81
2.2 下载图片56	2.7 简单公开文件服务器中的文档... 85
2.3 下载网页包含的所有图片61	2.8 构筑简易的记事本系统..... 91
2.4 文件上传72	2.9 构筑邮件送信的调查 表单 (Form) 96
2.5 动态生成静态 HTML 网页代码...76	

第 3 章 构筑简单的数据库应用程序

3.1 记录每页的浏览数101	3.4 构筑简单的网络调查程序 系统 117
3.2 图形化访问计数器107	3.5 下载表数据 125
3.3 在数据库中存储 Session 信息 ...111	

3.6 管理系列网络内容	129	3.8 提供 REST 形式的 API	135
3.7 动态生成网络书籍的目录	132	3.9 利用数据库保存图片	138

第 4 章 灵活应用 PEAR 与 Zend Framework

4.1 Zend_Mail: 构筑邮件送信 表单	143	4.10 JpGraph——由数字数据动态 生成图表	182
4.2 Zend_Mail: 通过 Web 浏览器 接受邮件	147	4.11 PEAR::Text_CAPTCHA——实现 垃圾信息对策程序	185
4.3 PEAR::Mail_Queue 大量邮件 批量送信	151	4.12 Qrcode PERL & PHP scripts 动态生成 QR 代码	189
4.4 PEAR::Text_Highlighter 高亮度 显示代码	156	4.13 Zend_Http 实现引用通告 功能	193
4.5 Zend_Translate/Zend_Date: 构筑支持多语言的网页	160	4.14 Zend_Service_Amazon 利用 Amazon Web 服务实现畅销书 的表示	196
4.6 FPDF/PFI: 动态生成 PDF 形式 的报表	164	4.15 PEAR::HTML_AJAX 利用 Google Maps API 实现地图 检索系统	201
4.7 PEAR::SpreadSheet_Excel_Writer: 根据数据库数据动态生成 Excel 报表	169	4.16 Smarty——应用模板引擎 更有效地管理外观	210
4.8 FeedCreator: 以 RSS 的形式发 布网页更新信息	173	4.17 Zend_Controller/Zend_View—— 在 PHP 中构筑 MVC 模式的 应用程序	214
4.9 MagpieRss 导入其他网站公开 的 RSS	178		

第 5 章 用 Zend Framework 与 Smarty 构筑真正的 MVC 模式应用

5.1 MVC 设计模式简介	224	5.5 共通功能——使用 Zend_Validate 进行输入数据的检查	238
5.2 系统分析	227	5.6 共通功能——使用 master 页 统一管理页面外观	247
5.3 共通功能: 使用 Zend_Db/Zend_ Config 建立数据库连接的共 同界面	229	5.7 共通功能——由 Zend_Auth 实现的用户认证功能	259
5.4 共通功能: 使用 JavaScript 进行 输入数据的检查	232		

5.8 共通功能——统一管理各个页面的标题与关键字 (SEO 对策)	274	5.16 书签共享系统——实现标签云 (PEAR::HTML_TagCloud)	351
5.9 共通功能——由 PEAR::HTML_Menu&Zend_Cache 实现的动态清晰的网站导航菜单	277	5.17 书签共享系统——个人朋友列表	356
5.10 共通功能——由 Zend_Log 实现的共同错误提示页与履历输出功能	290	5.18 书签共享系统——书签检索 ...	363
5.11 共通功能——利用别名, 隐藏网页真实路径	294	5.19 书签共享系统——修改书签系统设置	371
5.12 共享书签系统——概要	299	5.20 书签共享系统——首页、书签分类一览及其 RSS Feed	385
5.13 共享书签系统——追加新书签	301	5.21 站内留言系统概要	400
5.14 书签共享系统——个人收藏夹的管理	318	5.22 站内留言系统——写成并发送留言	402
5.15 书签共享系统——书签详细	337	5.23 站内留言系统——留言一览 (收件箱、草稿箱、送件箱、垃圾箱)	409
		5.24 站内留言系统——留言检索 ...	417

附录 A 文件系统函数

A.1 路径函数	423
----------------	-----

附录 B 网络函数篇

B.1 HTTP 关联函数	460
B.2 Session 处理函数	463
B.3 URL 函数	465

附录 C 数据操作函数

C.1 PDO 函数	467
C.2 SimpleXML 函数	475
C.3 XMLReader 函数	481
C.4 XMLWriter 函数	486
C.5 XSL 函数	488
C.6 Image 函数	490

附录 D 其他函数

D.1 选项与信息.....	504
D.2 JSON 函数.....	509
D.3 程序执行函数.....	510
D.4 错误处理/履历记录函数.....	511
D.4 其他函数.....	514

附录 E 设置文件

附录 F Smarty

F.1 Smarty 类.....	525
F.2 变量的调节符.....	531
F.3 模板函数.....	534

索引.....	544
---------	-----

第 1 章

导言

什么是 PHP？听说用 PHP 进行编程好像挺简单的，它和其他语言有什么区别呢？什么是服务器端技术？什么又是客户端技术呢？在学习应用 PHP 进行编程前，不可避免有许多诸如此类的疑问，本章在介绍具体的例子之前，先试着帮助大家解决这些疑惑。

粗看本章，肯定有不少怀着诸如“实际的编程时，哪用得着这些啰里啰嗦的介绍呀”抱怨的朋友。请先稍安勿躁，对于一名程序员（Programmer）或系统工程师（System Engineer）来说，适当了解关于技术发展的由来，以及技术未来的发展方向，肯定对自身的职业发展是有帮助的。

本章中，首先介绍了各种 Web 技术的兴衰轨迹，以及 PHP 在其中的位置，或许能帮你在未来给用户提案时，能够很好地回答“为什么要选择 PHP”的问题。同时还介绍了 PHP+MySQL 的环境配置，后面几节的例子将用到这个环境。如果自己配置出来的环境不能正确运行，请务必参照本章的配置。

1.1 PHP 技术的由来

1.1.1 服务器端脚本的发展轨迹——服务器端技术以前的技术

Web 技术的发展总是曲线的，有时因为某种原因而得到修正提高，各种各样的技术兴盛而又衰退。复杂的技术进步轨迹也是 Web 发展的一个很有趣的地方。从这个意义上说，服务器技术不仅是 Web 系统的最佳解决方案，而是 Web 技术发展的一个重要方面。

本节将介绍 PHP（PHP，Hypertext Preprocessor）所代替的服务器端技术在整个 Web 技术发展中所处的位置，以及 PHP 在服务器端技术中所处的分量。

HTML（HyperText Markup Language）

在急速的文档电子化的潮流中，曾出现过各种各样格式的电子文档。可是做成的电子文档，只是实现精美印刷这样的一个目的。很多被称为“标准”的格式，最终并没有成为标准，而且现在很多都消失了。

Web 是作为基础技术以打破文档电子化时代混乱的面貌出现的，广泛应用的 HTML 技术是其中的标准技术之一。HTML (HyperText Markup Language) 里有诸如<HTML>、<h1>的“对 (tag)”。在“对”中间包含具体的文字，作为标示语言文档的构造，格式的定制都非常简单。尽管有许多像语义不够精确、实际安装时文件比较大等不足的地方，但是 HTML 作为 Web 系统的基础技术之一，功绩是不可磨灭的。后来，随着插件技术 (Plugin)、CSS (Cascading Style Sheets) 技术等的发展，HTML 技术的表现力得到了飞跃的进步，但是最终 HTML 也只能提供生成静态文档的功能。因为那时，Web 的主要目的只是文档的公布。

ActiveX/Java Applet

后来，CSS 和 HTML 保持着长期共存的关系，插件技术并没有流行多久。确实曾经有一个时期，有很多各种各样的插件出现，给用户呈现了华丽而优美的网页效果。但是，因为需要强制用户在客户端安装插件程序，总是给人不够安全的感觉，不可能被大多数用户所接受。

这个时候，作为替代方案——ActiveX、Java 小程序 (Applet) 出现了。ActiveX 和插入的 Java 小程序直接嵌入在网页中，运行时自动安装所需要的控件，省略了插件导入的步骤。ActiveX 和 Java 小程序给开发人员带来了最早在客户端需要编程的想法。

网页内容有了动态的变化，确实给用户带来了享受的感觉，但是这两样技术的流行时间可以说都比较短。一个重要的原因就是用户需要掌握比 HTML 更高级的语言，如 Visual Basic、C++、Java 等语言，而用户能实现的只是“界面”等表面化的东西。当然，当时的网络环境，如网络速度、网络普及程度也不够发达，应用 ActiveX、Java 小程序的网页都显示出得太慢，超过了用户的忍耐度，也是其没能流行起来的其中一个原因。

客户端、服务器端脚本

但是，ActiveX 和 Java 小程序技术所带来的“客户端编程”的想法，在用户间流行起来了。显示的网页不再是静止不动，而是随着用户的输入、鼠标单击，网页显示的内容就会动态变化。所谓“交互式网站”的 Web 开发一时间流行起来了，这个时期正是全球规模的信息共享的目标达成之时，开发者的意识也从“如何实现信息共享”发展到了“如何做到与纸媒体的差别化”上来了。

能实现这些的就是客户端脚本了。以 Javascript、VBScript 为代表的客户端脚本语言是能在浏览器上运行的“简易”语言。它们本来就是为浏览器定制的，所以与 ActiveX 及 Java 小程序比较起来，运行时显得“轻”得多。基于这个理由，这两种脚本已被用户广泛接受了。后来又将 Dynamic HTML 及 DOM (Document Object Model) 的概念引进来，这样脚本语言的应用范围大大地扩展，功能也强化了。

但是，这些用脚本语言开发出来的程序网页尽管看起来是动态的，其实数据资源是静态的。因此，客户端、服务器端脚本是 Web 上“文字公开”的解决方案，并不是 Web 的基本构架。

1.1.2 服务器端脚本的发展轨迹——CGI 的时代

Web 是在独特的文化中成长起来的，项目的前提有时也会变成了最终目的。正如 1.1 节里介绍的那样，Web 本来的目的就是为了“文档公开共享”，但是从计算机技术的观点来看，要达成这个目的，需要解决浩如烟海的信息的输入/输出，也伴随着输入/输出的数据流程的问题。Web 只有输出的部分，在某种意义上显得极端不自然。

客户端脚本的极限

客户端脚本在 Web 的数据处理中，有各种各样的极限，下面将客户端脚本内包含的问题一一列举出来。

- 不能信息加工

客户端脚本只是实现了表面的动态化（Dynamic），所谓 Dynamic HTML，只是从表面上表述了客户端脚本的特性。而本质上，客户端脚本并不能改变网页上的内容，从这个意义上讲还没有逃脱“文档公开”这个范围。

- 逻辑的复杂性与网页大小的比例

客户端脚本，顾名思义，它只能在客户端（浏览器）上运行，必须先将所有数据都下载到客户端才能执行。如果网页有 10 000 条数据，那么首先就要将这 10 000 条数据下载到客户端，而网页实际上用到的可能只是其中很少的几条，其余都是不需要的，这样一来就影响了网页的显示速度。这种抽取需要或不需要数据的功能，是客户端脚本所不具备的。

- 客户端环境依存

这里提醒大家的一点是，说到客户端，并不是只指桌面个人电脑的客户端，手机以及其他手持电子设备也包括在内，形态是多种多样的。

对于依存客户端环境的客户端脚本来说，随着终端设备不同，对应不同的编程。从发展到现在的情况来看，开发者将所有终端设备的情况都考虑进去进行编程设计是不可能的。造成的结果就是在客户端只能安装各终端设备共通的功能，这样客户端脚本的适用范围就大大变小了。

- 程序的安全性

如“密码”这种机密性很高的信息在网页中处理的情况是很常见的。使用客户端脚本时，必须全在客户端公开。一些必须隐藏的信息，也必须全公开。最近出现的 encode/unicode 技术，在一定程度上解决了这些问题。

CGI (Common Gateway Interface)

以上客户端脚本所面对的问题，其实是数据处理都是在客户端进行这个原因造成的，只要数据不在客户端处理时，这些问题就迎刃而解。那么，客户端以外的处理环境就是服务器了。

服务器上存储了文档、程序，以及其他相关数据。那些对用户来说没有用处的程序、不需要的数据其实不用通过网络传送了，需要隐藏的数据就不会通过网络泄露。服务器将完成所有的处理，只是将处理的结果以 HTML 的形式传送到客户端的浏览器。

这些服务器端的处理模式，其实并不陌生。Perl (Practical Extraction and Reporting Language) 和以 C 为代表的 CGI (Common Gateway Interface)，就是以这些现在大家都熟悉的模式运行的 (如图 1-1 所示)。

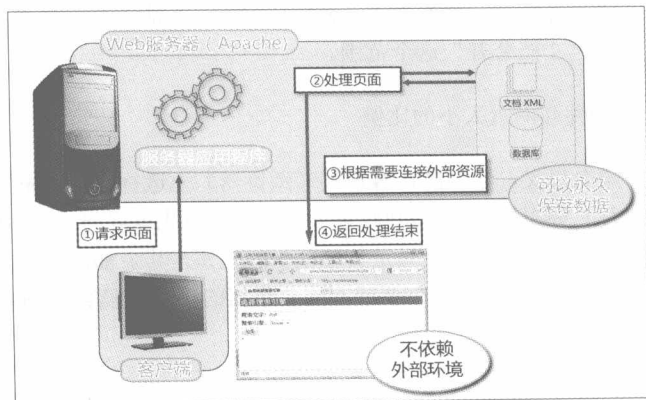


图 1-1 服务器端脚本的运作机制

在内容为主体的网页中，以开发网页的计数器、访问列表这样的小功能为目的，CGI 具体对于以数据处理为中心的服务器端技术来说，缺点是显而易见的。CGI 在开发难度、保守性、性能表现来讲，都有不足之处。

- 耗费服务资源太多

CGI 是从服务器上调用的外部程序的总称，也就是说，用户每发出一次请求 (Request)

就需要建立一个新的执行进程，用户送出多少次的请求就要生成多少次通信进程。对小的 CGI 程序来说，对服务器资源的占用不会有什么问题，但对那些复杂的商业应用程序来讲，有限的服务器资源就有被耗尽的危险，这就是所谓的“资源瓶颈”问题。

对于用户数变动很大的 Web 系统来讲，资源变动的脆弱性这个缺点可以说是致命的。

- 开发时要注意底层的协议

Perl 和 C 语言并不是专门为 Web 程序而开发的，CGI 开发者在开发中，必须要注意照顾到 Web 特有的协议（如 UTP，HTTP 等）。这些不仅仅意味着掌握 CGI 编程需要一定的技术水平，同时也意味着 CGI 存在开发瓶颈。

Perl 开发的程序里没有“Session”的概念，A 网页上的信息经过 B 网页，准备在 C 网页上显示时，CGI 开发者必须自行设计框架来解决这个问题。

其他的，连现在 Web 开发中经常使用的“Cookie”，在 CGI 中也必须将 HTTP 考虑进来专门编写代码。还有，当开发者想将一系列的程序当成一个项目来组织时，CGI 就很难办到了。

- 网页设计与编程开发

Perl 也好，C 也罢，实际上是通用开发语言，只是将处理结果按照 HTTP 的规则显示出来而已。所以如果就连一个 HTML 的“对”都得用 print 语言将其输出出来的话，则这就意味着，对于开发周期短的 Web 开发来说，所有作业都得在其中编程。而逻辑由程序员负责，内容由编辑负责，设计由设计人员来分担的这种合作开发模式，在 CGI 开发中就很难办到了。

1.1.3 服务器端脚本轨迹——服务器端脚本

正因为 CGI 有以上所介绍的这些缺点，慢慢地就不怎么受开发者的青睐了。这时候就出现了被称为 CGI 后继的技术。PHP 就是这些技术之一。下面就 PHP 是一种什么样的技术，有些什么样的特性做些简单的介绍。

- 脚本语言

PHP 是一门初级程序员都可以用来写出有用有效程序的语言，有着逐行顺次进行解释的、很容易重复 and-error 式的循环，能保证很高的生产效率等优势。性能表现尽管不能和那些能编译的语言相比，但是从 PHP4 开始，搭载上高性能的脚本处理引擎 Zend Engine 后，也能实现高速的处理，实际运行时几乎不产生瓶颈问题。在最新的 PHP5 中，Zend Engine 升级到 2.0 后，处理的效率更高了。

- HTML 嵌入式语言

PHP 必须嵌入在 HTML 中才能运行。与 HTML 有天然的亲和性，只要处理则能写出很优秀的程序。结果直接在 HTML 中输出就可以了，不需要输出部分的编程，实际的逻辑部分非常短小精悍。

- 容易掌握的高级语言

与 Perl、Javascript 这些 C 语言系列的语言很相似，PHP 也拥有自己的语言格式。掌握了上述语言的开发人员很容易上手，即使是第一次接触脚本语言的程序员，基本上都能在短时间内掌握它。

现在，PHP 中面向对象的思想也得到了强化，对于重复使用性、维护性好的大规模网站也能很容易地搭建了。

- 丰富的程序库

毫不夸张地说，要是想实现的功能，都能找得到相应的 PHP 程序库。PHP 开发时，将单个功能的程序块进行组合，就能实现复杂的商业逻辑，不用从零开始写那些原始的逻辑了。也正因为这样，初级程序员就能很容易地读懂 PHP 程序。

- 容易与各种类型的数据库连接

像 MySQL、PostgreSQL 等免费的数据库，还有 Oracle、Sybase、Informix、SQL Server 等商用的数据库，以及 ODBC 等通用的数据库接口等主要的数据库产品接口都内置在 PHP 程序包中，且可以自由安装或卸载。

最新的 PHP5 中轻量数据库引擎 SQLite 也已被内置。只要安装了 PHP，就能构筑数据库连接的应用程序了。当然，数据库连接和服务器端技术并非等价，但肯定是个极其重要的优势。PHP 和数据库的无缝结合能提供更简便、高级的解决方案。

1.1.4 各种服务器技术的比较

上述介绍的 PHP 构架并不是新鲜的技术。这之前介绍的以 Perl、C 等为代表的古典的 CGI，以及 Windows 平台的 ASP.NET，还有以跨多平台的处理环境为特征的，由 Java 语言构筑的 JSP/Servlet 等，都有很广泛的支持。

就服务器端脚本技术来讲，各自的技术特征总是千差万别的。本节选择其中几种典型脚本语言与 PHP 进行比较，让大家从横向上更好地了解一下 PHP。

CGI 与 PHP

正如以前所介绍的，CGI 是古典的服务器端技术，在开发的生产性、可维护性、易学性上不怎么让人满意。

然而，PHP 本来就是为制作 Web 网站而定制的语言，功能也是为 Web 定制的。那些经常使用的功能都作为函数提前准备好了，入门者可以很容易学会。在 PHP 中出错信息的生成功能也是很充分的，在程序调试时可以充分感受到。

因为 PHP 已经和 Web 服务器组装在了一起，因此不用每次生成请求时都生成新的进程了。虽然在 Web 服务器里组装进了 PHP 脚本翻译器而运行时显得“重”，但随着现在 Web 服务器中引进了动态模块的概念后，可根据需要动态载入，这在一定程度上解决了这个问题。

ASP.NET 与 PHP

ASP.NET 是在 Windows 环境下运行的服务器端脚本，在 HTML 中嵌入被称为服务器控件（Server Control）的“标签”（tag），拥有在服务器端技术中被称为事件驱动模型的独特构架。

在 ASP 3.0 中抛弃了以前的 VBScript、JScript 等解释型语言，采用了 Visual Basic、C# 等编译型语言后，成功地提升了性能，并提供了各种语言都能使用的类库。灵活的数据库应用、强大的文件处理、字符串处理、简便的自定义函数等，都是其用户迅速增加的原因。

但是，就其运行环境限制在 Windows+IIS（Internet Information Services）平台的这一点来讲，在普及上还是有很大的局限性的。最近由 Mono 工程组（<http://www.mono-project.com>）开发的能在 Linux 等 UNIX 系列操作系统运行的，开源码的 .NET 环境的出现，给普及带来了一丝希望。从实际使用情况来看，在 UNIX 环境下广泛使用 .NET 还有很长的路要走。

然而，PHP 原则上在 UNIX/Windows/Mac 等操作系统上都能运行，从 Web 服务器产品来说，有 Web 服务器最大市场占有率第一的 Apache，还支持 IIS、Sun Java System Web 服务器等重要的服务器产品，平台的通用性正是 PHP 的魅力之处。

JSP/Servlet 与 PHP

JSP（Java Server Pages）/Servlet 是具有代表性的 Web 开发语言 Java 的服务器端版本。

大家知道 Java 语言追求“Write Once, Run Anywhere”的开发概念，这在多样化的服务器环境中是非常重要的因素。JSP/Servlet 不用选择环境，其代码在不同的环境中能保证很好的移植性。Servlet 负责应用程序的逻辑部分，而 JSP 用于网页显示，彼此分工明确。

逻辑部分能够被再利用，从而能保证很高的生产效率，以及可维护性。

但是，Java 语言再怎么好用，与脚本语言比较起来，对一般的开发人员来说，还是显得门槛有点高。尽管最近标榜着 EoD (Ease of Development 容易开发)，以及比如 JSF (Java Server Faces)、Groovy 等新技术也纷纷登场，与追求直观易懂的 PHP 比较起来，还是需要花更多的时间去学习的。

JSP 的分工开发的优势，对那些中小规模的网站来说，优势显得并不明显，然而，PHP 在构筑中小规模的网站时拥有更大的灵活性。最新的 PHP5.0 中强化了面向对象的概念后，构筑大规模的网站时，开发再利用性好，可维护性强的代码成为可能。

表 1-1 总结了各种服务器技术与 PHP 的比较。

表 1-1 各种服务器技术与 PHP 的比较

技术	优点	缺点
CGI	开发语言可自由选择	入门的门槛高
	完全移植选择的语言的优点	不支持 Session 等
ASP.NET	较高的性能	运行环境限定
	类库/控件丰富	
	与开发环境联系紧密	
JSP/Servlet	不可自由选择运行环境	入门的门槛高
	再利用性/可维护性好	不适合小规模的系统
	较高的性能	
PHP	入门的门槛低	必须学习新语言 PHP
	可伸缩性较好	
	不可自由选择运行环境	
	也适合小规模系统开发	
	支持面向对象编程（再利用性好）	

1.2 PHP 的发展

前面在 PHP 与其他各种技术的比较中，介绍了 PHP 的优缺点。下面，从 PHP 的发展变迁的轨迹来论述一下 PHP 技术。

PHP 1.0

PHP 与其说是一门独特的脚本语言，还不如说是用 Perl 写成的 CGI 包装程序。后来，为了提高性能又用 C 语言进行了重写。这就是 1994 年发布的 PHP 1.0。这个时候，也只是解释