

PHP+MySQL

项目开发与实践

王爱华 著

PHP+MySQL

山东人民出版社

（第一版）

PHP+MySQL

PHP+MySQL

项目开发与实践

王爱华 著

山东人民出版社

国家一级出版社 全国百佳图书出版单位

图书在版编目(CIP)数据

PHP + MySQL 项目开发与实践/王爱华著. —济南:
山东人民出版社, 2014. 10

ISBN 978 - 7 - 209 - 08591 - 5

I. ①P… II. ①王… III. ①PHP 语言—程序设计 ②关
系数据库系统 IV. ①TP312 ②TP311.138

中国版本图书馆 CIP 数据核字 (2014) 第 233232 号

责任编辑: 李岱岩

PHP + MySQL 项目开发与实践

王爱华 著

山东出版传媒股份有限公司

山东人民出版社出版发行

社 址: 济南市经九路胜利大街 39 号 邮 编: 250001

网 址: <http://www.sd-book.com.cn>

发行部: (0531) 82098027 82098028

新华书店经销

莱芜市华立印务有限公司印装

规 格 16 开 (184mm × 260mm)

印 张 18.5

字 数 370 千字

版 次 2014 年 10 月第 1 版

印 次 2014 年 10 月第 1 次

ISBN 978 - 7 - 209 - 08591 - 5

定 价 39.80 元

如有质量问题, 请与印刷厂调换。电话: (0634 - 6216033)

前言

近年来,PHP 已经成为流行的 web 开发语言之一,作为一种功能强大的 web 编程语言,PHP 以其简单易学、安全性高和跨平台等特性而受到广大 web 开发者的喜爱,很多网站建设中都使用了 PHP+MySQL 技术,各网站开发公司需要大量这方面的人才,很多学校也都将 PHP+MySQL 作为动态网站编程课程中要讲解的核心技术。

随着越来越多的人开始学习使用 PHP,相关的教材也曾出不穷。但是传统的教材在编写中多偏重于 PHP+MySQL 的基础知识的讲解,通常的做法是一个知识点配以一个小练习,整本教材看起来更像是一本使用手册,知识结构是松散的,针对知识点设计的实训内容也是松散的,难以用完整的教学项目将这些知识点贯穿成一个整体。而且,几乎所有这方面的教材在编写时都是只谈 PHP,与网站界面设计中的 HTML、CSS 和 JavaScript 没有关联,读者通过书本学习的知识都是零碎的,因此在培养读者实践能力方面很难形成一个完整的知识体系,而在实际开发一个网站时,前台与后台、静态与动态往往是密不可分的。

鉴于这样一些情况,撰写一本以项目贯穿知识,实用性强,将 PHP 的知识与 HTML、CSS 和 JavaScript 有效融合的教材就显得非常有必要。

笔者近几年在 PHP 的教学过程中,在充分考虑 PHP 知识在网站中的应用需求之后,模拟开发了 163 邮箱网站的主要功能,在反复使用和修改之后,沉淀出最适合于初学者学习使用和开发的项目,该项目中应用了 PHP 在 web 开发中的主流技术,有效融合了静态网站中的 HTML、CSS 和 JavaScript 技术,适合于在各大浏览器中运行。

本书在撰写过程中,大部分理论知识的介绍都是根据项目开发的需要引出的,内容选取上不追求泛泛而谈,而是以实用性为原则,突出了项目开发和案例教学,避免了空洞的描述,每学到一个新知识点,读者都会很清楚如下几个问题:这是什么、如何使用、用在哪里,真正做到学以致用。

本书分为基础篇、核心篇与提高篇三大部分,共 11 个任务。

“基础篇”共包含了 4 个教学任务,分别是 PHP 基础知识简介、PHP 环境搭建、PHP 语法基础和表单数据提交,主要是为项目开发准备一些基础知识。

“核心篇”共包含了 5 个教学任务,分别是邮箱注册功能实现、邮箱登录功能实现、

邮箱写邮件功能实现、接收阅读和删除邮件功能实现以及在线投票与网站计数功能实现。详细讲解了开发过程中需要使用的各种技术,包含数据提交、文件上传、图片验证码的创建及应用、session 机制的应用、数据库的连接、分页浏览以及文件访问操作等动态网站开发中的核心技术,说明了在开发过程中需要注意的各种事项,并提供了全部完整的代码。

“提高篇”共设计了2个教学任务,分别是注册表单的复杂数据验证和复杂附件的添加与处理方法,这一部分内容可作为广大网站设计爱好者的选学模块。

全书内容组织按照循序渐进的原则展开,内容详细实用,旨在培养读者开发实际网站的能力,适合作为高等职业院校计算机类学生的专业课教材,也适合应用PHP开发动态网站的人员学习使用。

目 录

CONTENTS

前 言 / 001

第一部分 基础篇

任务一 PHP 基础知识简介 / 003

1.1 静态网页与动态网页的工作原理 / 003

1.1.1 静态网页与工作原理 / 003

1.1.2 动态网页与工作原理 / 005

1.2 初识 PHP / 006

1.3 习 题 / 007

任务二 PHP 程序的运行环境搭建 / 009

2.1 配置 Apache 服务器 / 010

2.1.1 安装 Apache 服务器 / 010

2.1.2 Apache 服务器安装过程中的问题及解决方案 / 012

2.1.3 Apache 主目录 / 014

2.2 安装 PHP 软件 / 015

2.2.1 安装 PHP 软件 / 015

2.2.2 修改 PHP 配置文件 / 017

2.3 安装 MySQL 数据库 / 018

2.3.1 安装 MySQL 数据库 / 018

2.3.2 正确卸载 MySQL 数据库 / 021

2.3.3 安装 MySQL 图形化工具 / 022

- 2.4 PHP 程序的开发工具 / 022
 - 2.4.1 安装及配置 Zend Studio for Eclipse / 023
 - 2.4.2 重置 Zend Studio 默认设置 / 025
- 2.5 习 题 / 026

任务三 PHP5 的基本语法 / 028

- 3.1 PHP 语法基础 / 028
 - 3.1.1 第一个 PHP 程序 / 028
 - 3.1.2 PHP 代码定界符与注释 / 030
 - 3.1.3 PHP 中的变量 / 030
 - 3.1.4 PHP 中的运算符 / 031
 - 3.1.5 PHP 程序的输出语句 / 031
- 3.2 流程控制语句 / 033
 - 3.2.1 常用的分支语句 / 033
 - 3.2.2 常用的循环语句 / 036
 - 3.2.3 日期时间函数 date() / 037
- 3.3 数 组 / 038
 - 3.3.1 PHP 数组的基本概念 / 038
 - 3.3.2 数组元素下标的用法 / 038
 - 3.3.3 使用 each() 函数遍历数组 / 040
- 3.4 数组及日期时间函数综合应用小示例 / 042
- 3.5 习 题 / 043

任务四 表单数据提交 / 045

- 4.1 表单界面设计及表单数据验证 / 046
 - 4.1.1 表单界面设计 / 046
 - 4.1.2 表单数据验证 / 050
- 4.2 表单数据提交 / 052
 - 4.2.1 系统内置数组 \$_POST 和 \$_GET / 053
 - 4.2.2 复选框组数据的提交 / 053
 - 4.2.3 获取并处理表单数据 / 054
 - 4.2.4 使用 isset() 函数解决单选按钮和复选框的问题 / 055
- 4.3 文件上传功能实现 / 056
 - 4.3.1 浏览器端的功能设置 / 056

- 4.3.2 服务器端的功能设置 / 056
- 4.3.3 简单文件上传小实例 / 057
- 4.3.4 上传并显示头像功能实现 / 059
- 4.4 习 题 / 061

第二部分 核心篇

任务五 163 邮箱注册功能实现 / 067

- 5.1 简单注册功能实现 / 068
 - 5.1.1 邮箱注册界面设计 / 068
 - 5.1.2 使用脚本验证注册数据 / 073
 - 5.1.3 服务器端获取并输出注册数据 / 074
- 5.2 使用图片验证码 / 075
 - 5.2.1 PHP 的图像处理函数 / 075
 - 5.2.2 创建图片验证码 / 077
 - 5.2.3 图片验证码的插入与刷新 / 082
 - 5.2.4 session 机制的原理与应用 / 083
 - 5.2.5 实现图片验证码的验证功能 / 087
 - 5.2.6 在 PHP 中引用外部文件 / 090
- 5.3 操作 MySQL 数据库 / 091
 - 5.3.1 MySQL 数据库操作界面简介 / 091
 - 5.3.2 创建数据库和数据表 / 092
 - 5.3.3 数据库的导入与导出 / 095
 - 5.3.4 PHP 文件访问 MySQL 数据库 / 098
- 5.4 使用数据库保存注册信息 / 102
 - 5.4.1 保存注册信息 / 102
 - 5.4.2 md5() 函数加密 / 105
- 5.5 小 结 / 106
- 5.6 习 题 / 107

任务六 163 邮箱登录功能实现 / 110

- 6.1 设计登录界面 / 110
 - 6.1.1 设计普通的登录界面 / 110
 - 6.1.2 设计 tab 选项卡式登录界面 / 113

6.2 完成登录功能 / 119

6.3 习 题 / 121

任务七

163 邮箱写邮件功能实现 / 123

7.1 设计主窗口界面文件 / 124

7.1.1 设计顶部区域 / 125

7.1.2 设计左下部区域 / 128

7.1.3 设计右下部区域 / 130

7.1.4 email.php 的完整代码 / 135

7.2 实现写邮件页面功能 / 136

7.2.1 布局、样式及页面元素插入 / 136

7.2.2 实现脚本功能 / 142

7.2.3 完整的 writeemail.php 代码 / 149

7.3 添加附件功能的实现 / 151

7.3.1 界面设计 / 151

7.3.2 使用脚本实现多附件添加和删除附件的功能 / 155

7.4 发送邮件 / 158

7.4.1 创建数据表 emailmsg / 158

7.4.2 保存邮件信息 / 159

7.4.3 实现系统退信功能 / 162

7.4.4 storeemail.php 文件的完整代码 / 166

7.5 小 结 / 167

7.6 习 题 / 168

任务八

接收、阅读、删除邮件功能实现 / 170

8.1 分页浏览邮件 / 171

8.1.1 收邮件功能描述 / 171

8.1.2 用 \$_GET 接收 URL 附加数据 / 172

8.1.3 处理查询结果记录集中的记录 / 176

8.1.4 分页浏览邮件 / 178

8.1.5 分页浏览中的数据验证 / 192

8.1.6 receiveemail.css 和 receiveemail.php 的完整代码 / 193

8.2 打开并阅读邮件 / 197

8.2.1 打开并阅读邮件页面的布局结构及功能要求 / 198

8.2.2 字符串替换函数 / 199

- 8.2.3 完成打开并阅读邮件页面的功能设计 / 203
- 8.2.4 openemail.css 和 openemail.php 文件的完整代码 / 208
- 8.3 删除邮件 / 210
 - 8.3.1 将邮件放入已删除文件夹 / 210
 - 8.3.2 分页浏览已删除文件夹中的邮件 / 212
 - 8.3.3 彻底删除邮件 / 212
- 8.4 小 结 / 214
- 8.5 习 题 / 215

任务九 在线投票与网站计数功能实现 / 217

- 9.1 文件系统函数 / 218
 - 9.1.1 文件的打开与关闭 / 218
 - 9.1.2 文件的读取与写入 / 219
- 9.2 在线投票功能实现 / 221
 - 9.2.1 简单在线投票功能实现 / 222
 - 9.2.2 使用 session 禁止反复投票 / 227
 - 9.2.3 使用 cookie 禁止重复投票 / 229
- 9.3 网站计数器功能实现 / 230
- 9.4 习 题 / 233

第三部分 提高篇

任务十 注册表单的复杂数据验证 / 237

- 10.1 整个页面的样式代码和页面文件代码 / 237
 - 10.1.1 样式代码 / 238
 - 10.1.2 页面内容代码 / 238
- 10.2 邮件地址的验证 / 241
- 10.3 密码验证与密码强弱的判断 / 244
- 10.4 其他数据验证 / 248
 - 10.4.1 确认密码验证 / 248
 - 10.4.2 手机号验证 / 249
 - 10.4.3 提交表单数据时的验证 / 251

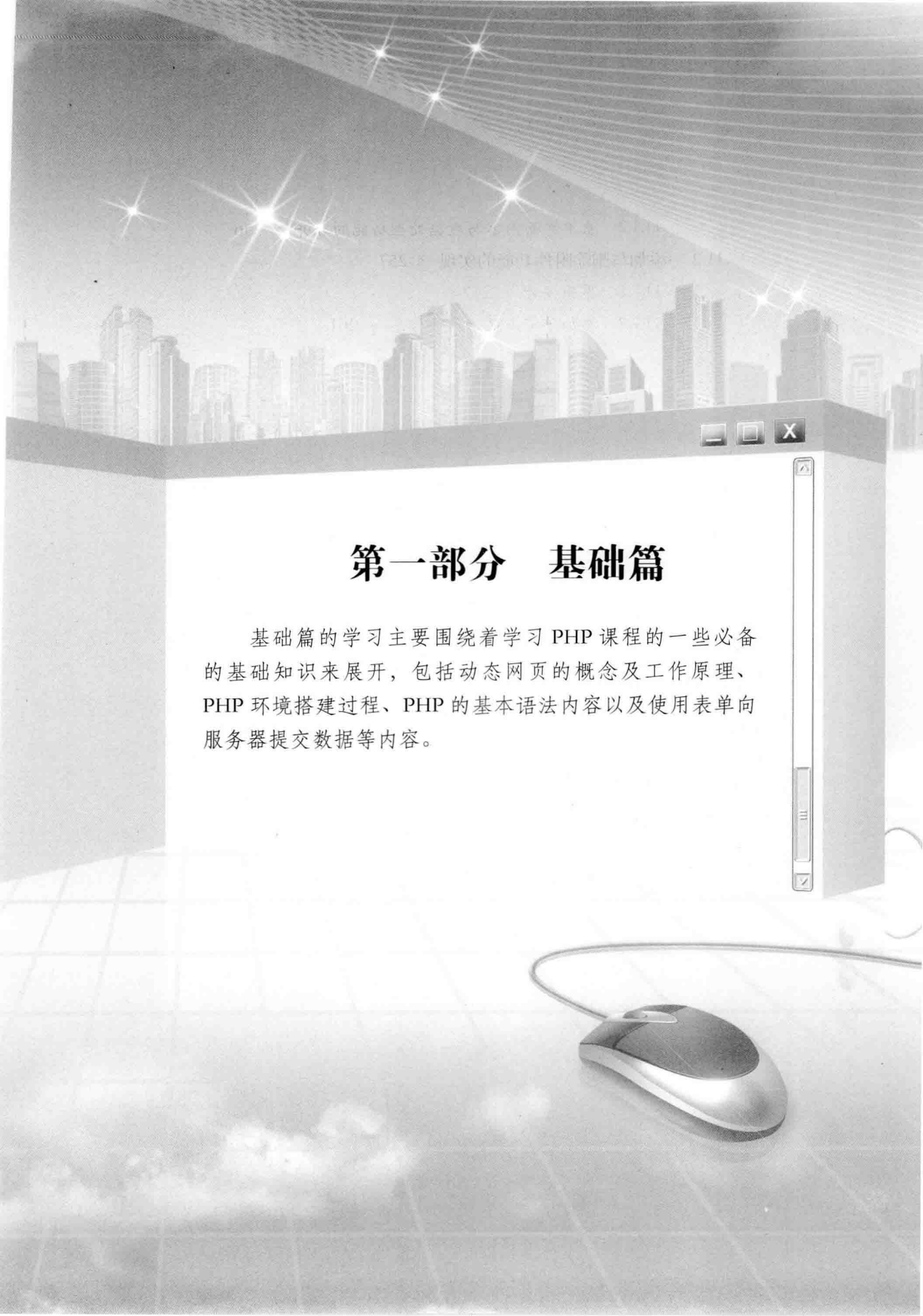
任务十一 复杂的附件添加与处理方法 / 253

- 11.1 设计“添加附件”页面 / 254
 - 11.1.1 选择附件的界面设计 / 255
 - 11.1.2 表单界面内容与数据处理功能的合并 / 256
- 11.2 添加与删除附件功能的实现 / 257
 - 11.2.1 界面设计 / 257
 - 11.2.2 添加子节点显示附件信息 / 261
 - 11.2.3 删除子节点以删除附件信息 / 266
- 11.3 存储和打开邮件的优化设计 / 268
 - 11.3.1 存储邮件的优化设计 / 268
 - 11.3.2 打开邮件界面的优化设计 / 269
- 11.4 几个常用的字符串操作函数 / 272
 - 11.4.1 字符串查找函数 / 272
 - 11.4.2 字符串截取函数 substr() / 273

附录 A PHP5 中的常用函数 / 275

- A.1 常用的数组操作函数 / 275
- A.2 常用的字符串操作函数 / 276
- A.3 常用的日期时间函数 / 278
- A.4 常用的数学函数 / 280
- A.5 文件系统函数 / 281
- A.6 MySQL 操作函数 / 282

附录 B 习题答案 / 285



第一部分 基础篇

基础篇的学习主要围绕着学习 PHP 课程的一些必备的基础知识来展开，包括动态网页的概念及工作原理、PHP 环境搭建过程、PHP 的基本语法内容以及使用表单向服务器提交数据等内容。



任务一

PHP 基础知识简介

主要知识点

- ① 静态网页与动态网页的区别
- ① 动态网页的工作原理
- ① PHP 的特性与功能说明



难 点

NAN DIAN

- ① 动态网页的工作原理
- ① PHP 的特性

1.1 静态网页与动态网页的工作原理

1.1.1 静态网页与工作原理

1. 静态网页

目前流行的静态页面都是用 HTML+CSS+JavaScript 编写的扩展名为 .htm 或 .html 的 HTML 文件。静态页面只能按一定格式显示固定的内容信息,无需用户提交信息,也无法按用户的要求显示任意需要的内容。

静态页面使用 JavaScript 可实现具有交互性的动态效果,例如动态变换图像、动

态更新日期、鼠标指向某个元素或区域时动态出现浮动区域、鼠标指向浮动区域内某个元素时还可动态改变显示效果并可实现超链接功能，但是，静态网页的所有动态内容都是事先设计好的固定内容，在页面刚刚运行时即下载到客户机器上，只不过是根据用户的操作由浏览器执行 JavaScript 产生动态效果而已，用户的操作及动态显示都与网站服务器没有关系，因此它并不是真正意义上的动态页面。

静态网页部署在服务器，收到客户请求后需要将整个页面的内容全部下载到客户机器上，由客户端浏览器运行。

静态网页最大的特点是，网页中显示的内容通常不会因人、因时的不同而不同，即任何人在任何时候来看页面时，内容都是一样的。

2. 静态网页的工作原理

当用户在浏览器地址栏中输入某网站的一个静态页面地址后，该网站服务器会通过 HTTP 协议把用户指定的“网页”文件及所有相关的资源文件传输到用户计算机中，再由用户计算机的浏览器解析执行该“网页”文件，将执行结果显示在浏览器中，从而形成用户看到的页面。

强调：静态页面一定是在浏览器端执行的。



图 1-1 静态网页的工作原理

静态网页的工作原理如图 1-1 所示。

静态页面的执行需要两步来完成：

第一步，在客户端浏览器地址栏输入 URL，向服务器发出 HTTP 请求；

第二步，服务器发回 HTTP 响应，将用户请求页面的所有代码及资源文件都返回给浏览器，浏览器解释执行之后，看到页面效果。

所以，我们访问的静态页面，在浏览器端查看源文件时，能够查看到文件的所有代码，不具有任何保密性。

1.1.2 动态网页与工作原理

1. 动态网页

动态页面的显著特点是，网页中显示的内容常常会因人、因时不同而不同。

例如我们登录邮箱，今天登录后阅读了几封未读邮件，明天登录后又会有新的未读邮件出现，即对同一个用户而言，不同时段看到的结果会有所不同，而不同的用户进入自己的邮箱看到的邮件更是大相径庭。动态网页可以按用户要求显示该用户保存在服务器上的信息，也可以将用户自己的信息提交给服务器保存。

再如微博网站，信息更是瞬息千变万化。

动态页面是用HTML+CSS+JavaScript结合ASP、PHP或JSP代码编写的.asp或.php或.jsp文件，即在一个动态页面代码中往往会穿插使用静态代码和动态代码。

动态页面的执行一般需要包含两个阶段：第一阶段，根据用户的请求在服务器端运行动态部分的代码，获取相应的结果；第二阶段，将动态代码获取的结果与源代码中的静态部分代码一起发送给浏览器，由浏览器解释执行这一部分内容，并将最后结果显示在浏览器界面上。

2. 动态网页工作原理

在动态页面中最常出现的功能是用户与服务器的交互，这种交互过程通常需要借助于表单界面，用户在表单界面中输入相关的信息之后，点击类似于“登录”、“注册”或“确认”等表单中的submit按钮，即可将数据提交给服务器，服务器端执行相关的动态页面文件，将结果回送到浏览器供用户浏览。

注意：并不是所有的动态页面都要借助于表单界面来提交数据，像在线投票、网站计数器以及一些简单的函数应用相对会比较简单一些。

这里直接以PHP程序的运行原理来说明动态网页的工作原理，如图1-2所示。

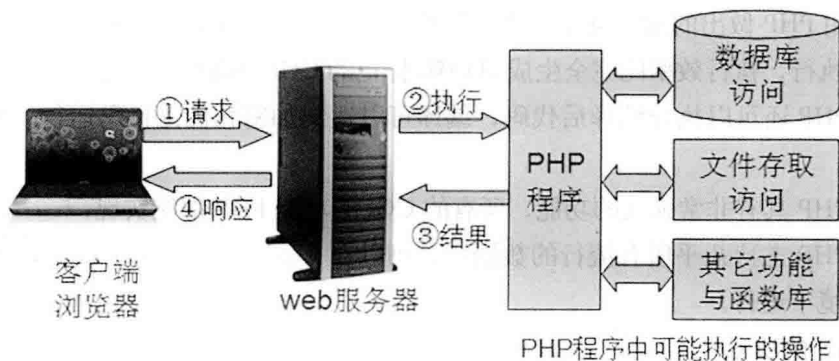


图 1-2 动态网页的工作原理

根据图 1-2 的流程可以看出, 动态网页的工作过程如下:

第一步, 由客户端浏览器使用 http 协议向 web 服务器发出请求, 请求运行的是包含了 PHP 代码的动态页面文件;

第二步, 由 web 服务器执行用户指定的 PHP 程序, 在程序执行过程中可能需要访问数据库, 可能要访问服务器端的文本文件, 也可能要执行其他功能或函数库中的函数;

第三步, web 服务器获取到 PHP 程序的执行结果;

第四步, web 服务器将 PHP 程序中动态部分代码的运行结果连同程序中原来的 HTML+CSS+JavaScript 部分的代码及需要的素材使用 HTTP 协议一起传送给客户端浏览器, 再由浏览器执行后以页面的形式展示给发出请求的用户。

到此, 一次访问过程结束。

强调: 动态页面的代码一定是在服务器上执行的。

1.2 初识 PHP

1. 什么是 PHP

PHP 的全称是 Personal HomePage: Hypertext Preprocessor (个人主页: 超文本预处理器)。

PHP 是一种服务器端的脚本语言, 必须在服务器环境下才能运行。它是开源免费和跨平台的, 而且具有高效安全和简单等特点, 使用 PHP 可写出功能强大的服务器端脚本。

2. PHP 的特性

- (1) PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创新的语法;
- (2) PHP 可以比 CGI 或者 Perl 更快速地执行动态网页;
- (3) 用 PHP 做出的动态页面与其他的编程语言相比, PHP 是将程序嵌入到 HTML 文档中去执行, 执行效率比完全生成 HTML 标记的 CGI 要高许多;
- (4) PHP 还可以执行编译后代码, 编译可以达到加密和优化代码运行, 使代码运行更快;
- (5) PHP 具有非常强大的功能, 所有的 CGI 的功能 PHP 都能实现;
- (6) PHP 支持几乎所有流行的数据库, 可以在 Linux、Unix 和 Windows 等各种操作系统环境下运行。

3. PHP 的功能

PHP 可以完成 web 服务器端的很多功能, 例如: