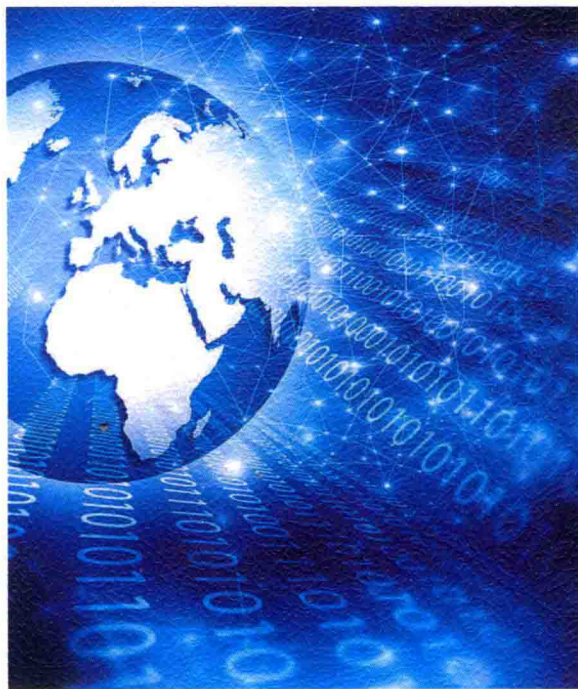


PHP+MySQL

动态网站开发基础教程

- ◆ 安装和配置PHP运行环境
- ◆ PHP语法、函数、字符串、数组
- ◆ 正则表达式的使用
- ◆ Web页面和PHP程序的交互
- ◆ 日期和时间功能，HTTP控制
- ◆ Cookie和Session
- ◆ PHP对MySQL数据库的操纵
- ◆ PHP对文件与目录的操纵
- ◆ GD库和图像技术
- ◆ PHP面向对象编程
- ◆ JavaScript和Ajax技术
- ◆ ThinkPHP框架的使用



李颖 编著



清华大学出版社

高等学校计算机应用规划教材

PHP+MySQL 动态网站 开发基础教程

李颖 编著



清华大学出版社

北京

内容简介

本书全面讲述了 PHP 程序设计语言的基本原理和技术。全书共分为 16 章, 深入介绍了 PHP 语言的用途与 PHP 环境的搭建、PHP 语法和函数、字符串、数组、正则表达式、PHP 与 Web 页面的交互、日期和时间、HTTP、Cookie 和 Session、数据库编程、用 PHP 操作 MySQL 数据库、文件操作、图像技术、面向对象编程、PHP 与 JavaScript 和 Ajax、ThinkPHP 等内容。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实, 主要面向 PHP 语言的初学者, 适合作为各种 Web 应用开发培训机构的培训教材、高等院校的 PHP 语言教材, 还可作为 Web 应用开发人员的参考资料。

本书的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

PHP+MySQL 动态网站开发基础教程/李颖 编著. —北京: 清华大学出版社, 2018
(高等学校计算机应用规划教材)
ISBN 978-7-302-50609-6

I. ①P… II. ①李… III. ①PHP 语言—程序设计—高等学校—教材 ②SQL 语言—程序设计—高等学校—教材 IV. ①TP312.8 ②TP311.132.3

中国版本图书馆 CIP 数据核字(2018)第 151305 号

责任编辑: 胡辰浩 李维杰

封面设计: 孔祥峰

版式设计: 思创景点

责任校对: 牛艳敏

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 23.75

字 数: 578 千字

版 次: 2018 年 7 月第 1 版

印 次: 2018 年 7 月第 1 次印刷

定 价: 59.00 元

产品编号: 074754-01

前言

本书将向读者介绍如何用 PHP 语言建立交互式的 Web 站点和应用程序。PHP 是当前最受欢迎的 Web 编程语言之一。作为编程语言，PHP 非常易学。PHP 提供了数以百计的内置函数，并且通过 PHP 引擎的增件，提供了数以千计的函数。本书只是把这门语言的最重要部分介绍给读者，指导读者创建可靠的、高质量的 PHP 应用程序。

本书是针对零基础编程学习者编写的 PHP 入门教程，从初学者角度出发，通过通俗易懂的语言、实用的示例，详细介绍了使用 PHP 进行程序开发所需要掌握的知识和技术。全书共分 16 章，内容包括开发环境的搭建、PHP 语言基础、流程控制语句、字符串操作与正则表达式、PHP 数组、面向对象编程、PHP 与 Web 页面的交互、MySQL 数据库基础以及综合网站开发等。书中的重要知识都结合具体实例进行讲解，对设计的程序代码也给出了详细注释，可以使读者轻松领会 PHP 程序开发的精髓，快速提高开发技能。

下面将逐章介绍本书的内容，这有助于读者决定采用什么样的阅读方法。

第 1 章为 PHP 入门。该章介绍 Web 技术基础、PHP 基础知识、PHP 开发环境的搭建、如何编写 PHP 程序、如何运行 PHP 程序、如何调试 PHP 程序和发生错误时如何处理等。

第 2 章介绍 PHP 语法和函数，主要内容包括 PHP 语法风格、标识符和关键字、常量、变量、变量类型的转换、运算符和表达式、流程控制语句、函数等。

第 3 章介绍字符串处理，主要内容包括创建和访问字符串、求字符串的长度、搜索字符串、字符串替换、格式化字符串、大小写转换等。

第 4 章介绍数组的知识，主要内容包括数组的概念、数组的创建和访问、数组的输出和切割、统计数组个数、遍历数组、多维数组的使用、数组的常用操作等。

第 5 章介绍正则表达式，首先介绍什么是正则表达式，其次讲解正则表达式的语法规则、模式匹配、正则表达式的使用、用 `preg_match_all()` 函数实现多次匹配、用 `preg_grep()` 函数搜索数组、用正则表达式实现文本替换功能等。

第 6 章介绍如何使用 PHP 建立交互式 Web 表单，具体内容包括如何建立 HTML 表单，如何在 PHP 脚本中捕获表单数据，如何用 PHP 生成动态表单。此外，还介绍文件的上传等。

第 7 章介绍常用的日期和时间功能以及 HTTP 请求，首先详细介绍日期和时间处理，包括许多实用的内置日期函数以及 `DateTime` 类，然后介绍如何处理 HTTP 请求-响应过程中的请求头和响应头等。

第 8 章介绍 Cookie 和 Session 技术，首先介绍 Cookie 的概念，Cookie 变量的创建、读取、删除，以及 Cookie 变量的生命周期，其次介绍 Session 的概念，Session 变量的创建，设置

Session 的有效时间,使用 Session 对用户权限进行控制,删除和销毁 Session 变量;最后介绍 Session 和 Cookie 的区别等。

第 9 章讨论 MySQL 控制台下的 MySQL 数据库编程,主要内容包括 MySQL 数据库的启动和连接、MySQL 数据库操作、数据表操作、数据表记录的更新操作、数据库的备份和还原操作等。

第 10 章介绍如何使用 PHP 语言操纵 MySQL 数据库,主要内容包括 MySQL 服务器的连接、SQL 语言的执行、数据库记录的读取、如何使用 PHP 语言对数据库执行增删改查操作等。

第 11 章介绍文件操作,首先介绍文件和目录基础、获取文件相关的属性信息、打开和关闭文件、文件的读写操作、文件权限的设置,其次介绍文件的赋值、重命名和删除,最后介绍目录的操作等内容。

第 12 章介绍如何使用 PHP 创建和处理图像,首先介绍计算机图形的一些理论,其次介绍如何创建新的图像,如何修改现有的图像,最后介绍如何使用 PHP 做一些常用的图像处理操作。

第 13 章主要介绍 PHP 面向对象技术,讨论如何使用 PHP 语言实现面向对象技术中的类、对象、类中的任何成员和属性、类中的静态成员、方法重载、接口、继承等内容。

第 14 章主要介绍 JavaScript 和 Ajax 技术与 PHP 语言的交互,主要内容包括 JavaScript 语言基础和 Ajax 技术的使用。

第 15 章主要介绍流行的 ThinkPHP 框架,重点介绍的是 ThinkPHP 的安装、目录架构、MVC 概念在 ThinkPHP 中的体现、URL 和路由等内容。

第 16 章主要通过两个综合实例来对本书介绍的 PHP 知识进行巩固,第一个综合实例通过 PHP 原生语言来实现,第二个综合实例通过 ThinkPHP 框架来实现。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实,每一章的引言部分概述了该章的作用和内容。在每一章的正文中,结合所讲述的关键技术和难点,穿插了大量极富实用价值的示例。每一章末尾都安排了有针对性的思考题和练习题,思考题有助于读者巩固所学的基本概念,练习题有助于培养读者的实际动手能力、增强对基本概念的理解和实际应用能力。

本书是专为 PHP 编程初学者编写的。如果读者以前使用过其他编程语言,如 Java、C# 或 Perl,则学习 PHP 会比较容易。本书适合作为高等院校 PHP 网站开发、Web 应用程序开发课程的教材,也可作为 Web 开发人员的参考资料。

全书由吉林师范大学的李颖编写并统稿。除封面署名的作者外,参加本书编写的人员还有周爱萍、屈文斌、万鑫、张春辉、梅泉滔、杨永好、郑梦成、孙红胜、何玉华、李文静、冯波、马金帅、张晓晗、张梦甜、李亮等。由于作者水平有限,本书难免有不足之处,欢迎广大读者批评指正。我们的电子邮箱是 huchenhao@263.net,电话是 010-62796045。

本书的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

作者

2018年5月

目 录

第 1 章 PHP 入门	1	2.1.2 PHP 注释	27
1.1 Web 技术基础	1	2.2 PHP 标识符与关键字	28
1.1.1 Web 技术概述	1	2.2.1 PHP 标识符	28
1.1.2 主流的 Web 应用平台	3	2.2.2 关键字	28
1.1.3 Web 工作原理	3	2.3 PHP 常量	29
1.1.4 常用的 Web 技术	4	2.3.1 常量的定义	29
1.2 PHP 基础知识	7	2.3.2 预定义常量	30
1.2.1 PHP 概述	7	2.4 PHP 变量	32
1.2.2 PHP 的优势	7	2.4.1 变量的声明和使用	32
1.2.3 PHP 的应用领域	8	2.4.2 PHP 的数据类型	33
1.2.4 常用的 PHP 开发工具	8	2.4.3 检测变量的数据类型	34
1.2.5 如何学好 PHP	8	2.4.4 可变变量	35
1.2.6 PHP 学习资源	9	2.4.5 变量的作用域	35
1.3 PHP 开发环境的搭建	12	2.5 变量类型的转换	37
1.3.1 WAMP 环境的搭建	12	2.5.1 自动类型转换	38
1.3.2 LAMP 环境的搭建	17	2.5.2 强制类型转换	38
1.3.3 扩展库	19	2.6 PHP 运算符与表达式	39
1.4 第一个 PHP 程序	21	2.6.1 运算符的类型	40
1.5 调试与错误处理	22	2.6.2 运算符的优先级	44
1.5.1 使用自带的报错功能	22	2.7 流程控制语句	45
1.5.2 引进调试工具	23	2.7.1 选择结构	45
1.5.3 调试业务逻辑错误	24	2.7.2 循环结构	49
1.5.4 调试非功能性错误	24	2.8 函数	53
1.6 本章小结	24	2.8.1 定义和调用函数	53
1.7 思考和练习	25	2.8.2 在函数间传递参数	54
第 2 章 PHP 语法和函数	26	2.8.3 从函数中返回值	55
2.1 PHP 的语法风格	26	2.8.4 变量函数	56
2.1.1 PHP 标记	26	2.8.5 对函数的引用	56
		2.8.6 取消引用	57

2.9	本章小结	58	4.2	数组的创建与访问	76
2.10	思考和练习	58	4.2.1	创建数组	76
第3章	字符串	59	4.2.2	访问数组的元素	76
3.1	创建和访问字符串	59	4.2.3	修改元素值	77
3.1.1	创建字符串	59	4.2.4	新增数组元素	77
3.1.2	在字符串中引用变量	60	4.2.5	输出所有数组元素	78
3.1.3	自定义分隔符	61	4.2.6	切割数组	79
3.1.4	求字符串的长度和单词 个数	62	4.2.7	统计数组中元素的个数	80
3.1.5	访问字符串中的字符	62	4.2.8	逐个访问数组的元素	80
3.2	搜索字符串	63	4.3	用 foreach 遍历数组	83
3.2.1	用 strstr()函数搜索字符串	63	4.3.1	用 foreach 遍历数组的 每个值	83
3.2.2	用 strpos()和 strrpos()函数 定位字符串位置	63	4.3.2	用 foreach 遍历数组的键 和值	83
3.2.3	用 substr_count()函数统计 字符串的出现次数	64	4.3.3	用 foreach 遍历修改数组 元素的值	84
3.2.4	用 strpbrk()函数搜索 字符集	65	4.4	多维数组	85
3.3	在字符串中进行字符替换	65	4.4.1	创建多维数组	85
3.3.1	用 str_replace()函数替换 全部搜索字符串	65	4.4.2	访问多维数组的元素	86
3.3.2	用 substr_replace()替换字符串 的部分内容	65	4.4.3	多维数组的遍历	86
3.3.3	用 strtr()函数变换字符	66	4.5	数组的操作	87
3.4	格式化字符串	66	4.5.1	数组排序	88
3.4.1	通用的格式化函数 printf()和 sprintf()	66	4.5.2	添加和删除数组元素	92
3.4.2	删除空白符的函数 trim()、 ltrim()和 rtrim()	70	4.5.3	合并数组	96
3.4.3	填充字符串函数 str_pad()	71	4.5.4	数组与字符串之间的转换	97
3.4.4	自动换行函数 wordwrap()	71	4.5.5	把数组转换为变量列表	98
3.4.5	格式化数值函数 number_format()	72	4.6	本章小结	99
3.5	字母大小写转换	73	4.7	思考和练习	99
3.6	本章小结	74	第5章	正则表达式	101
3.7	思考和练习	74	5.1	什么是正则表达式	101
第4章	数组	75	5.1.1	正则表达式的概念	101
4.1	数组概述	75	5.1.2	正则表达式的使用场景	102
			5.2	正则表达式的语法规则	103
			5.2.1	行定位符(^和\$)	103
			5.2.2	单词定界符(\b、\B)	103
			5.2.3	字符类([])	103
			5.2.4	选择字符()	104
			5.2.5	连字符(-)	104

- 5.2.6 排除字符([\^])..... 104
- 5.2.7 限定符(?*+{n,m})..... 104
- 5.2.8 点字符(.)..... 105
- 5.2.9 转义字符(\)..... 105
- 5.2.10 反斜线(\)..... 106
- 5.2.11 圆括号字符(())..... 107
- 5.2.12 反向引用..... 107
- 5.3 PHP 中的模式匹配..... 107
- 5.4 正则表达式的使用..... 108
 - 5.4.1 匹配字面字符..... 108
 - 5.4.2 用字符类匹配字符类型..... 109
 - 5.4.3 多次匹配同一字符..... 110
 - 5.4.4 贪婪匹配法和非贪婪匹配法..... 110
 - 5.4.5 用子模式分组模式..... 110
 - 5.4.6 引用前面的子模式匹配串..... 111
 - 5.4.7 匹配多个模式..... 111
- 5.5 用 preg_match_all()函数实现多次匹配..... 111
- 5.6 用 preg_grep()函数搜索数组..... 115
- 5.7 文本替换..... 116
 - 5.7.1 用 preg_replace()替换文本..... 116
 - 5.7.2 用 preg_replace_callback()替换文本..... 117
- 5.8 本章小结..... 118
- 5.9 思考和练习..... 118
- 第 6 章 PHP 与 Web 页面的交互..... 119**
 - 6.1 表单..... 120
 - 6.1.1 创建表单..... 120
 - 6.1.2 表单元素..... 120
 - 6.2 将表单保存为 HTML 文件..... 123
 - 6.3 获取表单数据的两种方法..... 125
 - 6.3.1 通过 POST 方式提交表单..... 125
 - 6.3.2 通过 GET 方式提交表单..... 125
 - 6.4 PHP 参数传递的常用方法..... 126
 - 6.4.1 通过 \$_POST[]接收表单数据..... 126
 - 6.4.2 通过 \$_GET[]接收表单数据..... 126
 - 6.5 在网页中嵌入 PHP 脚本..... 127
 - 6.5.1 在 HTML 标记中添加 PHP 脚本..... 127
 - 6.5.2 对表单元素的 value 属性进行赋值..... 127
 - 6.6 在 PHP 中获取表单数据..... 127
 - 6.6.1 获取基本表单元素的值..... 128
 - 6.6.2 获取单选按钮的值..... 129
 - 6.6.3 获取复选框的值..... 130
 - 6.6.4 获取下拉列表框/菜单列表框的值..... 130
 - 6.6.5 获取文件域的值..... 131
 - 6.7 对 URL 传递的参数进行编码和解码..... 132
 - 6.7.1 对 URL 传递的参数进行编码..... 132
 - 6.7.2 对 URL 传递的参数进行解码..... 133
 - 6.8 Web 服务器端的其他数据采集方法..... 134
 - 6.8.1 预定义变量 \$_REQUEST..... 134
 - 6.8.2 预定义变量 \$_SERVER..... 134
 - 6.9 文件上传..... 136
 - 6.9.1 配置 php.ini 以实现 PHP 文件上传功能..... 136
 - 6.9.2 文件上传函数..... 137
 - 6.9.3 多文件上传..... 139
 - 6.10 本章小结..... 140
 - 6.11 思考和练习..... 140
- 第 7 章 日期和时间、HTTP..... 141**
 - 7.1 日期和时间的处理..... 141
 - 7.1.1 时间戳..... 141
 - 7.1.2 获取当前日期和时间..... 142
 - 7.1.3 创建时间戳..... 142

7.1.4	转换时间戳	144	9.1.2	数据库常见术语	175
7.1.5	格式化日期字符串	146	9.2	MySQL 的启动与连接	176
7.1.6	检查日期值	148	9.2.1	启动 MySQL 服务	176
7.1.7	毫秒的使用	148	9.2.2	连接和断开 MySQL 数据库	178
7.1.8	DataTime	150	9.2.3	配置 Path 环境变量	178
7.2	HTTP 的使用	152	9.3	操作 MySQL 数据库	179
7.2.1	HTTP 请求	152	9.3.1	创建和查看数据库	180
7.2.2	HTTP 响应	153	9.3.2	删除数据库	181
7.2.3	修改 HTTP 响应方式	155	9.3.3	选择数据库	181
7.3	本章小结	156	9.3.4	综合实例——数据库的创建和删除	181
7.4	思考和练习	156	9.4	操作数据表	184
第 8 章	Cookie 和 Session	157	9.4.1	创建数据表	184
8.1	管理 Cookie	157	9.4.2	查看表结构	186
8.1.1	了解 Cookie	157	9.4.3	修改表结构	187
8.1.2	创建 Cookie	159	9.4.4	重命名数据表	187
8.1.3	读取 Cookie	159	9.4.5	删除数据表	187
8.1.4	删除 Cookie	160	9.5	数据记录的更新操作	188
8.1.5	Cookie 的生命周期	161	9.5.1	添加数据记录	188
8.2	管理 Session	161	9.5.2	查询数据记录	188
8.2.1	了解 Session	161	9.5.3	修改数据记录	189
8.2.2	创建 Session	162	9.5.4	删除数据记录	189
8.2.3	设置 Session 的有效时间	163	9.6	MySQL 数据库的备份与还原	190
8.2.4	通过 Session 对用户操作权限进行判断	164	9.6.1	使用 MYSQLDUMP 命令备份数据库	190
8.2.5	删除和销毁 Session	167	9.6.2	使用 mysql 命令还原数据库	190
8.2.6	Session 和 Cookie 的区别	168	9.7	本章小结	191
8.2.7	Session 和 Cookie 的应用	169	9.8	思考和练习	191
8.3	Session 的高级应用	169	第 10 章	用 PHP 操作 MySQL 数据库	192
8.3.1	Session 临时文件	169	10.1	PHP 访问 MySQL 数据库的基本步骤	192
8.3.2	Session 缓存	169	10.2	PHP 操作 MySQL 数据库的方法	194
8.3.3	Session 自动回收	170	10.2.1	连接 MySQL 服务器	194
8.3.4	php.ini 中的 Session 配置	171	10.2.2	选择 MySQL 数据库	195
8.4	本章小结	172			
8.5	思考和练习	173			
第 9 章	数据库编程	174			
9.1	MySQL 简介	174			
9.1.1	客户端/服务器(Client/Server) 软件	174			

10.2.3	执行 SQL 语句	196	11.7.2	目录对象	227
10.2.4	mysql_fetch_array() 函数	197	11.7.3	区分文件和目录	227
10.2.5	mysql_fetch_object() 函数	199	11.8	本章小结	229
10.2.6	mysql_fetch_row()函数	199	11.9	思考和练习	229
10.2.7	mysql_num_rows()函数	200	第 12 章	图像技术	230
10.2.8	释放资源	201	12.1	计算机图形基础	230
10.2.9	关闭连接	201	12.1.1	色彩原理	230
10.3	PHP 操作数据库	202	12.1.2	坐标系	231
10.3.1	PHP 操作数据库	202	12.1.3	图像类型	231
10.3.2	动态添加用户信息	203	12.2	使用 GD 库	232
10.3.3	查询数据信息	205	12.2.1	GD&PHP 可以处理的 文件类型	232
10.3.4	修改数据	206	12.2.2	在 PHP 中启用 GD 库	232
10.3.5	删除数据	209	12.3	创建图像	233
10.4	本章小结	210	12.3.1	新建图像	234
10.5	思考和练习	210	12.3.2	颜色分配	234
第 11 章	文件操作	211	12.3.3	输出图像	235
11.1	文件与目录基础	211	12.3.4	在图像上进行绘制	236
11.2	获取文件的信息	212	12.4	处理图像	241
11.2.1	获取文件的时间属性	212	12.4.1	打开图像	241
11.2.2	从路径获取文件名	212	12.4.2	添加水印	242
11.3	打开和关闭文件	213	12.4.3	制作缩略图	245
11.3.1	用 fopen()打开文件	213	12.4.4	添加标准化文本	246
11.3.2	用 fclose()关闭文件	214	12.4.5	使用 TrueType 字体	247
11.4	文件的读写	214	12.5	本章小结	249
11.4.1	读写字符串	215	12.6	思考和练习	249
11.4.2	文件末尾的测试	217	第 13 章	面向对象编程	250
11.4.3	一次读取一行内容	218	13.1	面向对象编程介绍	250
11.4.4	读取 CSV 文件	218	13.2	创建类和对象	252
11.4.5	读取和写入整个文件	219	13.2.1	创建类	252
11.4.6	随机存取文件数据	221	13.2.2	类的成员	252
11.5	文件的权限	222	13.2.3	实例化类	254
11.5.1	改变文件的权限	222	13.3	创建和使用属性	255
11.5.2	检查文件权限	223	13.3.1	声明属性	255
11.6	文件的复制、重命名和删除	223	13.3.2	属性的可见性	255
11.7	目录	224	13.3.3	访问属性	256
11.7.1	其他目录函数	225	13.3.4	静态属性	257

13.3.5	类常量	257	14.3.2	循环语句	288
13.4	方法	258	14.3.3	跳转语句	290
13.4.1	创建方法	258	14.4	JavaScript 事件	291
13.4.2	方法的可见性	258	14.5	调用 JavaScript 脚本	291
13.4.3	方法的调用	258	14.5.1	在 HTML 中嵌入 JavaScript 脚本	291
13.4.4	方法的参数和返回值	259	14.5.2	应用 JavaScript 事件调用自定义函数	292
13.4.5	在方法中访问对象的属性	259	14.5.3	在 PHP 动态网页中引用 JS 文件	292
13.4.6	静态方法	261	14.6	在 PHP 中调用 JavaScript	293
13.5	用 __get()、__set()和 __call()重载对象	261	14.6.1	使用 JavaScript 脚本验证表单元素是否为空	293
13.5.1	用 __get()和 __set()方法重载属性访问	262	14.6.2	使用 JavaScript 脚本制作二级导航菜单	294
13.5.2	用 __call()重载方法调用	263	14.6.3	使用 JavaScript 脚本控制文本域和复选框	297
13.5.3	其他重载方法	265	14.7	Ajax 技术	298
13.6	继承与接口	266	14.7.1	Ajax 的概念	298
13.6.1	重载父类的方法	268	14.7.2	Ajax 的开发模式	298
13.6.2	保留父类的功能	270	14.7.3	Ajax 的优点	299
13.6.3	用 final 类和方法阻止继承和重载	270	14.7.4	XMLHttpRequest	300
13.6.4	抽象类和抽象方法	271	14.7.5	在 Ajax 开发过程中需要注意的问题	302
13.6.5	接口	273	14.7.6	用户重名检测	303
13.7	自动加载类文件	276	14.8	本章小结	305
13.8	序列化类对象	276	14.9	思考和练习	305
13.9	判断一个对象所属的类	278	第 15 章	ThinkPHP	306
13.10	本章小结	279	15.1	ThinkPHP 简介	306
13.11	思考和练习	280	15.1.1	ThinkPHP 的安装	307
第 14 章	PHP 与 JavaScript 和 Ajax	281	15.1.2	ThinkPHP 概述	308
14.1	了解 JavaScript	281	15.2	ThinkPHP 架构	309
14.1.1	什么是 JavaScript	281	15.2.1	ThinkPHP 的目录结构	310
14.1.2	JavaScript 的功能	282	15.2.2	自动生成项目目录	311
14.2	JavaScript 语言基础	282	15.2.3	命名规范	314
14.2.1	JavaScript 数据类型	282	15.2.4	资源目录	315
14.2.2	JavaScript 变量	283	15.2.5	调试模式配置	315
14.2.3	JavaScript 注释	284	15.2.6	控制器	315
14.3	JavaScript 流程控制语句	285			
14.3.1	条件语句	285			

15.2.7	视图	318	16.1.6	程序员完成网站功能实现	337
15.2.8	数据读取	319	16.1.7	网站上线测试	337
15.3	URL 和路由	320	16.1.8	网站推广	337
15.3.1	URL 访问	320	16.2	留言板	338
15.3.2	参数传入	321	16.2.1	留言板制作预备知识	338
15.3.3	隐藏入口	322	16.2.2	留言板功能需求分析	338
15.3.4	定义路由	322	16.2.3	留言板数据库表设计	339
15.3.5	URL 生成	323	16.2.4	留言信息的读取展示	339
15.4	请求与响应	323	16.2.5	留言表单及留言处理	341
15.4.1	请求对象	323	16.2.6	后台管理登录	344
15.4.2	请求信息	325	16.2.7	PHP 留言板系统后台管理	345
15.4.3	响应信息	326	16.2.8	后台管理回复及留言删除处理	346
15.5	数据库操作	327	16.3	个人博客	347
15.5.1	数据库配置	327	16.3.1	功能阐述	347
15.5.2	原生查询	327	16.3.2	功能结构	348
15.5.3	链式操作	328	16.3.3	系统预览	348
15.5.4	事务支持	328	16.3.4	数据库设计	349
15.6	模型和关联	329	16.3.5	数据表设计	349
15.6.1	模型定义	329	16.3.6	连接数据库	351
15.6.2	基础操作	330	16.3.7	自动生成项目目录	351
15.7	视图和模板	332	16.3.8	控制器的设置	352
15.7.1	模板输出	332	16.3.9	视图设置	355
15.7.2	分页输出	333	16.3.10	后台管理程序架构分析	360
15.7.3	渲染内容	334	16.3.11	数据模型和数据校验的设计	362
15.8	本章小结	334	16.4	本章小结	365
15.9	思考和练习	335	16.5	思考和练习	365
第 16 章	综合实例	336	参考文献		367
16.1	网站开发流程	336			
16.1.1	确定建站目标	336			
16.1.2	进行需求分析	336			
16.1.3	绘制网站原型	337			
16.1.4	系统整理所需资料	337			
16.1.5	与网站设计美工确定布局和风格	337			

第 1 章

PHP 入门

PHP 是一门服务器端程序设计语言。除 PHP 外，还有 JSP、ASP、ASP.NET 等重要的服务器端程序设计语言。服务器端程序语言主要运行在服务器端，用于处理来自浏览器端的客户请求；服务器端程序根据请求处理好之后，将处理结果返回到浏览器端，供用户在浏览器端查看或进行下一步交互。

PHP 语言最强大和最重要的特征就是跨平台和面向对象。本章首先介绍 Web 技术基础，其次介绍 PHP 语言的基础知识、开发环境的搭建，接着编写一个简单的 PHP 程序，最后介绍开发过程中遇到错误如何调试和处理。通过本章的学习，读者能够对 PHP 语言有一个整体上的认识，为后期学习 PHP 语言的具体内容打下良好的基础。

本章的学习目标：

- 掌握 Web 技术基础知识。
- 了解 PHP 基础知识。
- 掌握 Windows 与 Linux 操作系统平台上 PHP 开发环境的搭建。
- 掌握编写、运行和调试 PHP 程序的方法。

1.1 Web 技术基础

没有 Web 技术，就没有 PHP 的诞生。因此，在学习 PHP 之前，首先来了解 Web 技术相关的一些知识。

1.1.1 Web 技术概述

1. Web 的定义

百度百科对 Web 的定义是：Web 一般指 WWW(World Wide Web)，即全球广域网，也称为万维网，是一种基于超文本和 HTTP 的、全球性的、动态交互的、跨平台的分布式图形信息系统，是建立在 Internet 上的一种网络服务，为浏览者在 Internet 上查找和浏览信息提供了图形化的、易于访问的直观界面，其中的文档及超链接将 Internet 上的信息节点组织成一个互为关联的网状结构。简而言之，就是指互联网，人们通常说的“上网”就是访问互联网。

2. Web 的表现形式

Web 的表现形式有 3 种：超文本、超媒体、超文本传输协议。

- 超文本：百度百科将超文本定义为一种用户接口方式，用以显示文本以及与文本相关的内容。通俗易懂地说，主要是指链接到其他字段或文档的超文本链接(超链接)，允许浏览者从当前阅读位置直接跳转到超文本链接所指向的文字。一般具有这种特性的文档就是网页——用 HTML(HyperText Markup Language, 超文本标记语言)语言书写的文档。
- 超媒体：超级媒体的简称，是超文本和多媒体在信息浏览环境下的结合，使得通过网页不仅能从一段文本跳转到另一段文本，还可以播放一段声音、显示一个图形，甚至可以播放一段动画。由此可见，超媒体使网页变得丰富多彩。
- 超文本传输协议(HyperText Transfer Protocol, HTTP)是互联网上应用最为广泛的一种网络传输协议。浏览器和服务端之间的交互都是通过 HTTP 协议来实现的。

3. C/S 和 B/S 架构

C/S 和 B/S 架构是最流行的两种软件架构。

(1) C/S 架构

C/S(Client/Server)架构，即客户端/服务器架构。客户端包含一个或多个在用户计算机上运行的程序；而服务器有两种，一种是数据库服务器，客户端通过数据库连接访问数据库服务器上的数据；另一种是 Socket 服务器，服务器上的程序通过 Socket 与客户端的程序通信。

C/S 架构也可以看成胖客户端架构。因为客户端需要实现绝大多数的业务逻辑和界面展示功能。这种架构中，作为客户端的部分需要承受很大的压力，因为显示逻辑和事务处理都包含在其中，通过与数据库的交互(通常是 SQL 或存储过程的实现)来达到持久化数据，以此满足实际项目的需要。

C/S 架构的优点有：界面和操作可以很丰富；安全性能很容易得到保证；实现多层认证也不难，由于只有一层交互，因此响应速度较快。缺点是：适用面窄，通常用于局域网中；用户群固定，由于程序需要安装才可使用，因此不适合面向一些不可知的用户；维护成本高，只要升级，所有客户端的程序都需要改变。

(2) B/S 架构

B/S(Browser/Server)架构，即浏览器/服务器架构。Browser 指的是 Web 浏览器，Server 是指用某种语言编写的服务器端程序。在 B/S 架构中，业务逻辑处理一般很少在浏览器端实现，主要放在服务器端用服务器端程序语言(后端语言)实现。一般情况下，浏览器、服务器和数据库构成了网站开发的三层架构。采用 B/S 架构的系统不需要特别安装客户端组件，用浏览器执行即可。

B/S 架构中，显示逻辑(即网页)交给 Web 浏览器解释执行，业务逻辑放在服务器端，用后端语言编写程序来处理，这样减少了客户端浏览器的压力。由于客户端浏览器只需要负责页面呈现和用户交互，因此也被称为瘦客户端。

B/S 架构的优点是：客户端无须安装组件，有浏览器即可；B/S 架构可以直接放在 Internet 上，供多用户访问，交互性较强；B/S 架构无须升级多个客户端组件，更新服务器端程序即可。

B/S 架构的缺点是：在跨浏览器上，B/S 架构在呈现上不尽如人意，要达到 C/S 架构的呈现程度更难；在速度和安全性上需要花费巨大的成本，这是 B/S 架构的最大问题；客户端/服务器的交互是请求-响应模式，常常需要刷新页面。

1.1.2 主流的 Web 应用平台

动态网站服务器平台至少要包括：操作系统+Web 服务器+应用程序服务+数据库。好的动态网站服务器是由多方面因素决定的，如个人喜好、部署费用、安全机制等。目前主流的 3 种 Web 平台分别是 LAMP、J2EE 和 ASP.NET，它们的运行环境组合如下：

- LAMP: Linux+Apache+MySQL+PHP
- J2EE: UNIX+Tomcat+Oracle+JSP
- ASP.NET: Windows Server+IIS+SQL Server+ASP.NET

LAMP、J2EE 和 ASP.NET 平台各有优缺点，三者的比较如表 1-1 所示。

表 1-1 3 种 Web 平台的比较

性能比较	LAMP	J2EE	ASP.NET
运行速度	较快	快	快
开发速度	快	慢	快
运行损耗	一般	较小	较大
难易程度	简单	难	简单
运行平台	Linux/UNIX/Windows	绝大多数平台	Windows 平台
扩展性	好	好	较差
安全性	好	好	较差
应用程度	较广	较广	较广
建设成本	非常低	非常高	高

1.1.3 Web 工作原理

Web 应用程序采用的是 B/S 架构。Web 工作原理就是：B/S 架构模式下，Web 服务器如何接收用户通过浏览器发来的请求，如何处理这些请求，以及如何将处理的结果返回给浏览器呈现给用户查看(即进行下一步交互)的过程。下面以 Apache 和 PHP 为基础，详细介绍 Web 工作原理。

1. 当直接请求静态 HTML 页面时

当客户通过浏览器直接请求静态的 HTML 页面时，即请求的页面不带应用程序和数据库操作时，Web 服务器将根据访问地址，找到存放该页面的地址，然后将该页面直接返回给客户端浏览器，如图 1-1 所示。

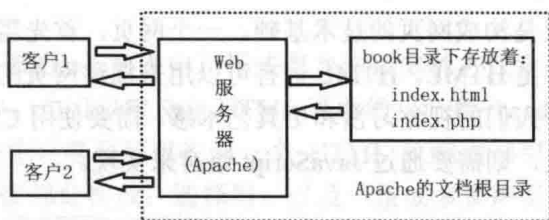


图 1-1 当直接请求静态 HTML 页面时

2. 当访问的页面带应用程序时

当客户请求一个带应用程序(如 PHP 程序)的页面时，Web 服务器将寻找到该文件，并通知

PHP 应用服务器。PHP 应用服务器(即 PHP 解释器)逐条解释程序, 将其翻译成 HTML 静态页面, 然后将该 HTML 静态页面返回给 Web 服务器, 由 Web 服务器返回给浏览器, 呈现给客户, 如图 1-2 所示。

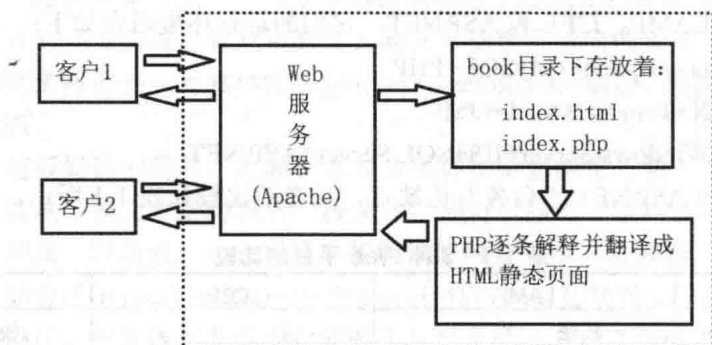


图 1-2 当访问带应用程序的页面时

3. 当访问的页面带应用程序和数据库时

当客户请求一个带应用程序(如 PHP 程序)的页面, 并且该页面需要访问数据库时, PHP 应用服务器逐条解释程序时, 还需要连接数据库服务器(如 MySQL 服务器), 并通过标准 SQL 语句来操作数据库, 得到结果, 返回给 PHP 程序, 翻译成 HTML 静态页面, 然后将该 HTML 静态页面返回给 Web 服务器, 由 Web 服务器返回给浏览器, 呈现给客户, 如图 1-3 所示。

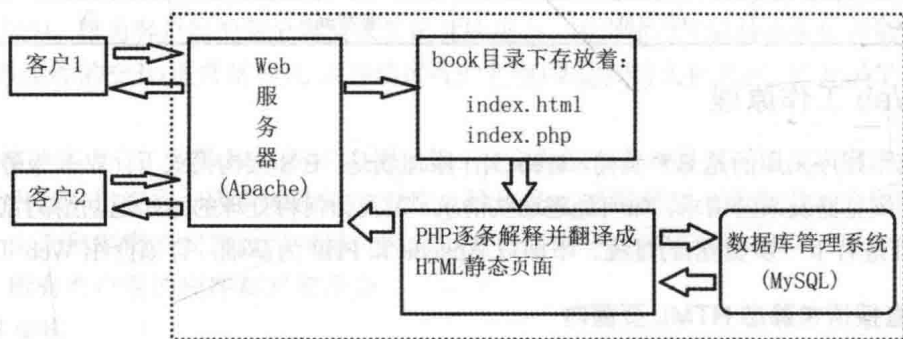


图 1-3 当访问的页面带应用程序和数据库时

1.1.4 常用的 Web 技术

Web 技术基础, 也就是构成网页的技术基础。一个网页, 首先需要有构成这个网页的结构。表示网页构成的语言是 HTML。HTML 语言可以用来描述网页的构成元素。为了使页面的布局或元素的样式符合人们的视觉习惯和更具艺术感, 需要使用 CSS 语言来描述。如果网页还要能和用户进行交互, 则需要通过 JavaScript 语言来实现。

1. HTML

HTML(HyperText Markup Language, 超文本标记语言)是用标记来描述网页的一种语言, 因此 HTML 不是编程语言, 而是标记语言。网站由一个个网页组成, 因此可以使用 HTML 来建立网站。HTML 网页运行在浏览器上, 由浏览器解析。

HTML 文件的扩展名为.html 或.htm。文件结构如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>HTML 文档标题</title>
</head>
<body>
  <h1>第一个标题</h1>
  <p>第一个段落</p>
</body>
</html>
```

其中，各组成部分的含义如下：

- <!DOCTYPE html>声明为 HTML 文档。
- <html>元素是 HTML 页面的根元素。
- <head>元素包含文档的元(meta)数据。
- <title>元素描述文档的标题。
- <body>元素包含可见的页面内容。
- <h1>元素定义一个大标题。
- <p>元素定义一个段落。

由此可见，HTML 语言主要由一个个标记来描述网页的组成及结构。标记格式为：<标记名></标记名>。由于本书主要讲解 PHP 语言，因此不对 HTML 作详细介绍，后面章节仅对使用到的标记进行介绍。若要深入了解 HTML 语言，可参考其他 HTML 教程。

2. CSS

HTML 标记主要用于定义网页结构，通过使用<h1>、<p>、<table> 这样的标记，HTML 表达的是“这是标题”“这是段落”“这是表格”之类的信息。同时，网页布局由浏览器来完成，而不使用任何的格式化标记。

由于浏览器不断地将新的 HTML 标记和属性(比如字体标签和颜色属性)添加到 HTML 规范中，创建文档内容清晰地独立于文档表现层的站点变得越来越困难。

为了解决这个问题，万维网联盟(W3C)肩负起了 HTML 标准化的使命，并在 HTML 4.0 之外创造出层叠样式表(CSS)。由此可见，CSS 主要是为了描述网页元素的呈现效果而诞生的，有了 CSS，HTML 可以专注于定义网页内容，CSS 则专注于定义 HTML 标记的呈现效果及页面布局。

样式通常保存在外部的.css 文件中。样式表允许以多种方式定义样式信息。样式可以定义在单个 HTML 标记中，也可以定义在 HTML 页面的头元素<head></head>中，还可以单独在一个外部的 CSS 文件中，甚至可以在同一个 HTML 文档内部引用多个外部样式表。

CSS 规则由两个主要部分构成：选择器，以及一条或多条声明。格式如下：

```
selector {declaration1; declaration2; ... declarationN }
```

其中，选择器(selector)通常是需要改变样式的 HTML 元素。每条声明(declaration)由一个属性(property)和一个值(value)组成。属性是希望设置的样式属性。每个属性有一个值。属性和值被冒号分开，格式如下：