

<<PHP和MySQL Web开发(原书)>>

图书基本信息

书名：<<PHP和MySQL Web开发(原书第4版)>>

13位ISBN编号：9787111262817

10位ISBN编号：7111262816

出版时间：2009

出版时间：机械工业出版社

作者：Luke Welling,Laura Thomson

页数：714

译者：武欣

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

内容概要

本书将PHP开发与MySQL应用相结合，分别对PHP和MySQL做了深入浅出的分析，不仅介绍PHP和MySQL的一般概念，而且对PHP和MySQL的Web应用做了较全面的阐述，并包括几个经典且实用的例子。

本书是第4版，经过了全面的更新、重写和扩展，包括PHP 5.3最新改进的特性（例如，更好的错误和异常处理），MySQL的存储过程和存储引擎，Ajax技术与Web 2.0以及Web应用需要注意的安全问题。

PHP和MySQL是非常流行的开源技术，它们非常适合快速开发数据库驱动的Web应用。PHP是一种功能强大的脚本语言，专门用于快速创建高性能的Web应用，而MySQL则是一个快速而又可靠的数据库，它能很好地与PHP集成，适用于基于互联网的动态应用。

本书介绍了如何使用这些工具创建高效和交互式的Web应用。它清晰地介绍了PHP语言的基础，解释了如何设置和使用MySQL数据，以及如何使用PHP与数据库和服务器进行交互。

本书非常实用，包括大量实际应用中的例子。例如，用户验证、创建购物车、动态生成PDF文档和图像、发送和管理电子邮件、管理用户讨论、使用XML连接Web服务，以及使用基于Ajax的交互性开发Web 2.0应用。

与上一版相比，本书经过了更新、重写以及扩展，并涵盖了PHP 5到5.3的所有特性，例如命名空间和闭包以及MySQL 5.1引入的特性。

作者简介

Luke Welling和Laura Thomson合作编写PHP和MySQL的图书已经超过了20年，如今他们还经常出现在一些开源会议上。

Luke Welling如今是OmniTI公司的Web架构师，而Laura Thomson是Mozilla公司Web团队的资深软件工程师。

书籍目录

读者反馈译者序前言第一篇 使用PHP 第1章 PHP快速入门教程 1.1 开始之前：了解PHP 1.2 创建一个示例应用：Bob汽车零部件商店 1.3 在HTML中嵌入PHP 1.4 添加动态内容 1.5 访问表单变量 1.6 理解标识符 1.7 检查变量类型 1.8 声明和使用常量 1.9 理解变量的作用域 1.10 使用操作符 1.11 计算表单总金额 1.12 理解操作符的优先级和结合性 1.13 使用可变函数 1.14 根据条件进行决策 1.15 通过迭代实现重复动作 1.16 从控制结构或脚本中跳出 1.17 使用可替换的控制结构语法 1.18 使用declare 1.19 下一章 第2章 数据的存储与检索 2.1 保存数据以便后期使用 2.2 存储和检索Bob的订单 2.3 文件处理 2.4 打开文件 2.5 写文件 2.6 关闭文件 2.7 读文件 2.8 使用其他有用的文件函数 2.9 文件锁定 2.10 更好的方式：数据库管理系统 2.11 进一步学习 2.12 下一章 第3章 使用数组 3.1 什么是数组 3.2 数字索引数组 3.3 使用不同索引的数组 3.4 数组操作符 3.5 多维数组 3.6 数组排序 3.7 多维数组的排序 3.8 对数组进行重新排序 3.9 从文件载入数组 3.10 执行其他的数组操作 3.11 进一步学习 3.12 下一章 第4章 字符串操作与正则表达式 4.1 创建一个示例应用程序：智能表单邮件 4.2 字符串的格式化 4.3 用字符串函数连接和分割字符串 4.4 字符串的比较 4.5 使用字符串函数匹配和替换子字符串 4.6 正则表达式的介绍 4.7 用正则表达式查找子字符串 4.8 使用正则表达式分割字符串 4.9 比较字符串函数和正则表达式函数 4.10 进一步学习 4.11 下一章 第5章 代码重用与函数编写 5.1 代码重用的好处 5.2 使用require()和include()函数 5.3 在PHP中使用函数 5.4 理解为什么要定义自己的函数 5.5 了解基本的函数结构 5.6 使用参数 5.7 理解作用域 5.8 参数的引用传递和值传递 5.9 使用Return关键字 5.10 实现递归 5.11 进一步学习 5.12 下一章 第6章 面向对象的PHP 6.1 理解面向对象的概念 6.2 在PHP中创建类、属性和操作 6.3 类的实例化 6.4 使用类的属性 6.5 使用private和public关键字控制访问 6.6 类操作的调用 6.7 在PHP中实现继承 6.8 类的设计 6.9 编写类代码 6.10 理解PHP面向对象新的高级功能 6.11 下一章 第7章 错误和异常处理 7.1 异常处理的概念 7.2 Exception类 7.3 用户自定义异常 7.4 Bob的汽车零部件商店应用程序的异常 7.5 异常和PHP的其他错误处理机制 7.6 进一步学习 7.7 下一章 第二篇 使用MySQL 第8章 设计Web数据库 8.1 关系数据库的概念 8.2 如何设计Web数据库 8.3 Web数据库架构 8.4 进一步学习 8.5 下一章 第9章 创建Web数据库 9.1 使用MySQL监视程序 9.2 登录到MySQL 9.3 创建数据库和用户 9.4 设置用户与权限 9.5 MySQL权限系统的介绍 9.6 创建一个Web用户 9.7 使用正确的数据库 9.8 创建数据库表 9.9 理解MySQL的标识符 9.10 选择列数据类型 9.11 进一步学习 9.12 下一章 第10章 使用MySQL数据库 10.1 SQL是什么 10.2 在数据库中插入数据 10.3 从数据库中获取数据 10.4 更新数据库记录 10.5 创建后修改表 10.6 删除数据库中的记录 10.7 表的删除 10.8 删除整个数据库 10.9 进一步学习 10.10 下一章 第11章 使用PHP从Web访问MySQL数据库 11.1 Web数据库架构的工作原理 11.2 从Web查询数据库的基本步骤 11.3 将新信息放入数据库 11.4 使用Prepared语句 11.5 使用PHP与数据库交互的其他接口 11.6 进一步学习 11.7 下一章 第12章 MySQL高级管理 12.1 深入理解权限系统 12.2 提高MySQL数据库的安全性 12.3 获取更多关于数据库的信息 12.4 数据库的优化 12.5 备份MySQL数据库 12.6 恢复MySQL数据库 12.7 实现复制 12.8 进一步学习 12.9 下一章 第13章 MySQL高级编程 13.1 LOAD DATA INFILE语句 13.2 存储引擎 13.3 事务 13.4 外键 13.5 存储过程 13.6 进一步学习 13.7 下一章 第三篇 电子商务与安全性 第14章 运营一个电子商务网站 14.1 我们要实现什么目标 14.2 考虑电子商务网站的类型 14.3 理解风险和威胁 14.4 选择一个策略 14.5 下一章 第15章 电子商务的安全问题 15.1 信息的重要程度 15.2 安全威胁 15.3 易用性、性能、成本和安全性 15.4 建立一个安全政策 15.5 身份验证原则 15.6 加密技术基础 15.7 数字证书 15.8 安全的Web服务器 15.9 审计与日志记录 15.10 防火墙 15.11 备份数据 15.12 自然环境安全 15.13 下一章 第16章 Web应用的安全 16.1 处理安全性问题的策略 16.2 识别所面临的威胁 16.3 了解与我们“打交道”的用户 16.4 代码的安全性 16.5 Web服务器和PHP的安全性 16.6 数据库服务器的安全性 16.7 保护网络 16.8 计算机和操作系统的的天性 16.9 灾难计划 16.10 下一章 第17章 使用PHP和MySQL实现身份验证 17.1 识别访问者 17.2 实现访问控制 17.3 使用基本身份验证 17.4 在PHP中使用基本身份验证 17.5 在Apache的.htaccess文件中使用基本身份验证 17.6 使用mod_auth_mysql身份验证 17.7 创建自定义身份验证 17.8 进一步学习 17.9 下一章 第18章 使

用PHP和MySQL实现安全事务 18.1 提供安全的事务处理 18.2 使用加密套接字层 (SSL) 18.3 屏蔽用户的输入 18.4 提供安全存储 18.5 存储信用卡号码 18.6 在PHP中使用加密技术 18.7 进一步学习 18.8 下一章 第四篇 PHP的高级技术 第19章 与文件系统和服务器的交互 19.1 文件上载 19.2 使用目录函数 19.3 与文件系统的交互 19.4 使用程序执行函数 19.5 与环境变量交互: getenv()和putenv() 19.6 进一步学习 19.7 下一章 第20章 使用网络函数和协议函数 20.1 了解可供使用的协议 20.2 发送和读取电子邮件 20.3 使用其他Web站点的数据 20.4 使用网络查找函数 20.5 备份或镜像一个文件 20.6 进一步学习 20.7 下一章 第21章 日期和时间的管理 第22章 创建图像 第23章 在PHP中使用会话控制 第24章 其他有用的特性 第五篇 创建实用的PHP和MySQL项目 第25章 在大型项目中使用PHP和MySQL 第26章 调试 第27章 建立用户身份验证机制和个性化设置 第28章 创建一个购物车 第29章 创建一个基于Web的电子邮件服务系统 第30章 创建一个邮件列表管理器 第31章 创建一个Web论坛 第32章 生成PDF格式的个性化文档 第33章 使用XML和SOAP来连接Web服务 第34章 使用Ajax构建Web 2.0应用附录 附录A 安装PHP及MySQL 附录B Web资源

章节摘录

5.1 代码重用的好处 软件工程师的一个目标就是通过重复使用代码来避免编写新的代码。这样做并不是因为他们懒，而是因为重新使用已有的代码可以降低成本、增加代码的可靠性并提高它们的一致性。

在理想情况下，一个新的项目是这样创建的：它将已有的可重新利用的组件进行组合，并将新的开发难度降低到最小。

5.1.1 成本 在一个软件的有效生命周期中，相当多的时间是用在维护、修改、测试和文档化记录上，而不是最初花在编码上的时间。

如果要编写商业代码，应该尽量限制结构中所用到的代码行数。

一个最常使用的方法就是：重新使用已有的代码，而不是为一个新任务编写一个和原来代码只有微小区别的新代码。

更少的代码意味着更低的成本。

如果市场上已经存在能够满足需求的软件，那就购买软件。

购买已有软件的成本总是要小于开发一个等价产品的成本。

如果有现成的软件基本上能够满足要求，那就必须小心地使用它。

修改已有的代码可能会比编写新代码更加困难。

5.1.2 可靠性 如果一个模块代码已经在代码结构中使用了，可以认为它是已经通过测试的。

即使代码只有几行，在重写时仍然可能忽略两方面的内容，一是原作者融入其中的某些东西，二是代码测试发现缺陷后，对原来代码添加的一些东西。

使用现存的成熟的代码通常要比新鲜的“绿色”代码更可靠。

5.1.3 一致性 系统的外部接口应该是一致的，其中包括用户接口和系统的外部接口。

编写一段新的并且能够和系统函数的其他部分保持一致的代码需要花些心思和努力。

如果重复使用运行在系统其他部分的代码，所实现的功能自然就会达到一致。

除了这些优点外，只要原来的代码是模块化的而且编写良好，那么重复使用代码还会节省许多工作。

在工作时，可以试着辨认一下今后可能再次要调用的代码段。

5.2 使用require()和include()函数 PHP提供了两个非常简单却很有用的语句，它们允许重新使用任何类型的代码。

使用一条require()或include()语句，可以将一个文件载入到PHP脚本中。

通常，这个文件可以包含任何希望在一个脚本中输入的内容，其中包括PHP语句、文本、HTML标记、PHP函数或PHP类。

这些语句的工作方式类似于大多数Web服务器提供的服务器端包含方式以及C语言或C++中的#include语句。

require()和include()几乎是相同的。

二者唯一的区别在于函数失败后，require()函数将给出一个致命的错误。

而include()只是给出一个警告。

require()和include()也有两个变体函数，分别是require_once()和include_once()。

正如你可能猜到的，这两个函数的作用是一个被引入（included）的文件只能被引入一次。

对于我们已经介绍过的例子——页眉和脚注（header and footer）——这个功能并不是非常有用。

当使用require()和include()来引入函数库时，它们才非常有用。

使用这两个函数可以防止错误的引入同样的函数库两次，从而出现重复定义的错误。

如果关心编码实践，可以考虑使用require()和include()，因为它们的运行速度较快。

5.2.1 文件扩展名和require()函数 如下所示的代码保存于reusable.php文件中：如下所示的代码保存于main.php文件中：

如果载入reusable.php，当浏览器中显示出“Here is a very simple PHP statement”时，你不会感到奇怪。

如果载入main.php，则会发生一件更有趣的事情。

该脚本输出结果如图5-1所示。

当需要一个文件的时候，可以使用require()语句。

在前面的例子中，我们使用的文件是reusable.php。

当运行该脚本时，require()语句：将被请求的文件内容代替，然后再执行脚本。

这就意味着，当载入main.php文件时，它会像如下所示的代码那样执行：当使用require()语句时，必须注意处理文件扩展名和PHP标记的不同方式。

PHP并不会查看所需文件的扩展名。

这就意味着，只要不想直接调用这个文件，就可以任意命名该文件。

当使用require()语句载入文件时，它会作为PHP文件的一部分被执行。

通常，如果PHP语句放在一个HTML文件（例如，名为page.html的文件）中时，它们是不会被处理的。

PHP通常用来解析扩展名被定义成.php的文件。

（在Web服务器配置文件中可能不是这样）但是，如果通过require()语句载入这个page.html，文件内的任何PHP命令都会被处理。

因此，可以使用任何扩展名来命名包含文件，但要尽量遵循一个约定，例如将扩展名命名为.inc是一个很好的办法。

需要注意的一个问题是，如果扩展名为.inc或一些其他的非标准扩展名的文件保存在Web文档树中，而且用户可以在浏览器中直接载入它们，用户将可以以普通文本的形式查看源代码，包括任何密码。

因此，将被包含文件保存在文档树之外，或使用标准的文件扩展名是非常重要的。

提示：在这个例子中，可重用文件（reusable.php）代码如下所示：我们将文件中的PHP代码放到PHP标记之间。

如果希望一个所需文件中的PHP代码能被当成PHP代码进行处理，就必须这样做。

如果不使用PHP标记，代码将会被视为文本或者HTML脚本，因此也就不会被执行。

5.2.2 使用require()制作Web站点的模版 如果Web页面具有一致的外观，可以在PHP中使用require()语句将模版和标准元素加到页面中。

例如，一个虚构的TLA咨询公司的网站有许多页面，这些页面的外观看上去都如图5-2所示。

当需要一个新页面的时候，开发人员可以打开一个已有页面，从文件中间剪切所需的文本，输入所需的新文本，然后以新的文件名保存。

考虑这种情况：网站已经存在了一段时间，如今已有数十个、数百个甚至数千个页面都是同一种风格。

现在，要对标准外观进行部分修改—这种修改可能是很微小的改变，例如，在每个脚注上加一个电子邮件地址，或者加上一个新的导航菜单入口。

你希望对这数十个、数百个甚至数千个页面都做这种微小的修改吗？

相对于剪切粘贴数十个、数百个甚至数千个页面，直接重用各个页面中通用的HTML代码部分是一个更好的办法。

程序清单5-1给出了图5-2所示页面(home.html)的源代码。

编辑推荐

《php和mysql web开发(原书第4版)》：开发人员专业技术丛书。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>