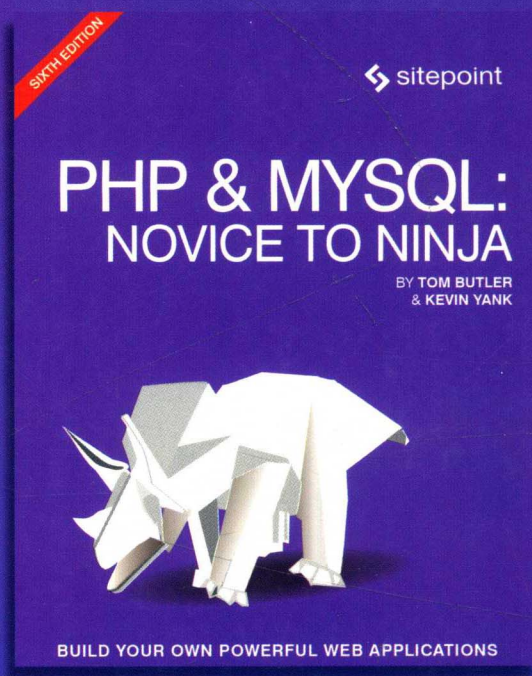


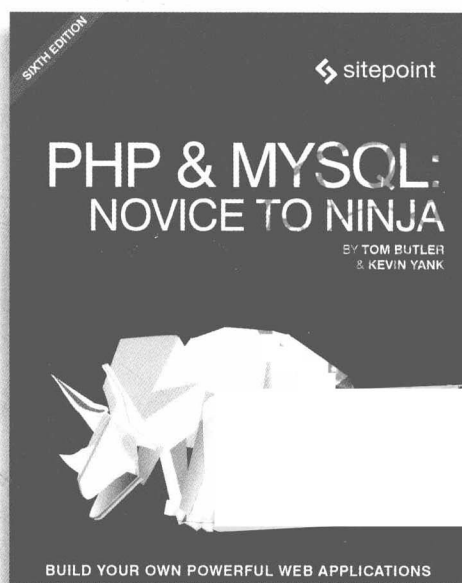
PHP 和 MySQL Web 开发学习指南 (第6版)

【澳】汤姆·巴特勒 (Tom Butler) 【澳】凯文·雅克 (Kevin Yank) 著
李强 译



PHP 和 MySQL Web 开发学习指南 (第6版)

【澳】汤姆·巴特勒 (Tom Butler) 【澳】凯文·雅克 (Kevin Yank) 著
李强 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

PHP和MySQL Web开发学习指南 / (澳) 汤姆·巴特勒
(Tom Butler), (澳) 凯文·雅克 (Kevin Yank) 著 ;
李强 译. -- 北京 : 人民邮电出版社, 2018. 12
ISBN 978-7-115-49369-9

I. ①P… II. ①汤… ②凯… ③李… III. ①PHP语言
—程序设计—指南②关系数据库系统—指南 IV.
①TP312.8②TP311.138

中国版本图书馆CIP数据核字(2018)第212558号

版权声明

Copyright © 2018 Posts & Telecom Press

Authorized Chinese Simplified translation of the English edition of *PHP & MySQL: Novice to Ninja, 6th Edition* (ISBN 978-0-9943469-8-8) © 2016 Sitepoint Pty. Ltd.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

本书由 **Sitepoint Pty. Ltd** 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

-
- ◆ 著 [澳] 汤姆·巴特勒 (Tom Butler)
[澳] 凯文·雅克 (Kevin Yank)
 - 译 李 强
 - 责任编辑 吴晋瑜
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
涿州市京南印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 22
字数: 706千字 2018年12月第1版
印数: 1—2400册 2018年12月河北第1次印刷
- 著作权合同登记号 图字: 01-2016-9210号
-

定价: 79.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

内 容 提 要

本书主要介绍构建现代 PHP Web 站点所需的技能、PHP 的基础知识以及现阶段开发者所使用的经过尝试和验证的技术。

本书共 14 章，从服务器和虚拟机的安装开始，介绍 PHP 和 MySQL 基础知识、在 Web 上发布 MySQL 数据、关系数据库设计、PHP 结构化编程、改进插入和更新函数以及对象和类等内容，然后在此基础上介绍如何创建一个可扩展的框架、如何进行 MySQL 管理以及如何用正则表达式进行内容格式化。

学完本书后，读者应能很好地理解 PHP，并能使所编写的代码更高效、更快速。本书要求读者掌握基本的 HTML 知识，适合从事服务器端编程的中级和高级 Web 设计师参考。

前 言

1998年，我12岁，那一年，父母给家里购买了第一台现代PC。没过多久，我就搞清楚了如何通过修改我所喜爱的第一人称射击游戏的代码来做一些小事情，例如让火箭弹发射器一秒发射100个火箭弹而不是一个火箭弹，然后，让它在每个方向上发射100个火箭弹，并且快速地搞垮游戏。我为此着迷，从此以后就开始编程。

这款游戏是多个玩家参与的。其他人也发现了修改代码的方法，并且“武装竞赛”很快升级。一些人朝我发射了100个火箭弹。我还编写了一个脚本，它能够准备好随时在我面前建起一堵墙来阻止对手的攻击。

对手在我的下方埋下了十几个地雷。我必须摆脱重力，然后跳起来，立即逃离即将发生的爆炸。如果每个人都能飞来飞去，这就变得没有什么乐趣了。你进入一个游戏中，某人却编写了一段脚本将你送到地图上的另一端，立即杀死你并且迫使你再生，一秒之内重复这个过程十多次。当然，他们还会冻结你的控制。

我发现了阻止所有这些的方式，但是，最终双方进入了僵持状态。试图首先进入游戏的任何人都将完全控制游戏，不管你的脚本有多么好，你什么也做不了。

这就是我学习编程基础知识的方式，而唯一的限制是你的想象力和创造力。在那个时候，我还自学了HTML，并且建立了自己的Web站点来分享一些游戏黑客技术和脚本。

到了2000年，我自学了PHP/MySQL基础知识并且运行针对一组程序员的Web站点。我编写了一些粗糙的PHP脚本在Web站点上发布新闻、进行问卷调查，甚至编写了一个脚本来处理小型锦标赛的排名和赛程安排。

此后，我转向使用一种“恐怖”的语言Delphi来编写桌面应用程序，编写工具帮助人们来改装各种游戏。2007年，我大学毕业并且拿到了软件工程的学位，在各种公司担任PHP开发者。在这些日子里，我回到大学拿到了博士学位，并开始作为一个演讲者来传播我对于编程的热情。

现在，我31岁了，并且已经在人生的很长一段时间内从事编程工作。这很有趣，有时候，我全身心地享受编程。我编写本书来分享我的知识，并帮助你清除那些容易掉进去的陷阱。学习编码是非常享受而又有成就感的事情。在编译自己的程序时，你可以看到它被赋予了生命。然而，这也可能是令人非常沮丧的体验。在本书中，我打算尝试用我自己的经验，带你走一段顺畅的旅程——要比我自己和很多程序员的旅程顺畅得多。我可以指导你起步，朝着正确的方向前进。

在介绍任何代码之前，我打算给你一些关于编程和学习编码的建议，这也是我给我的所有学生的建议。

读者对象

本书针对那些想要立即开始服务器端编程的中级和高级Web设计师。我期望你已熟悉简单的HTML，因为我肯定会使用HTML并且不会给出太多说明。我不会假设或要求你拥有层叠样式表(CSS)或JavaScript方面的知识，但是，如果你熟悉JavaScript，你会发现这使得学习PHP更加容易，因为这些语言很相似。

在学完本书之后，你将能够掌握构建现代PHP Web站点所需的技能、PHP的基础知识以及如今的开发者所使用的经过尝试和验证的技术。

编程已经发生了变化

和2001年时编写一个Web站点的那些程序员相比，现在才从头开始的新程序员需要学习比那时多得

多的知识，才能够发布一个 Web 站点。当我开始学习的时候，那是一个相当简单的时代。例如，Web 站点安全性并不是需要太多考虑的问题。除非你是一家银行或者从事信用卡支付的公司，很少会有人以你的站点为攻击目标。

然而如今，每一个 Web 站点都会受到自动程序和脚本的轰炸，而这些程序和脚本专门探测你可能留下的甚至是最小的后门。

编写 PHP 脚本的方式也发生了显著的变化，肯定是更好了。现在，很容易下载其他人的代码，并用在你自己的项目中。这一点的缺点是，你需要对编程概念有更加广泛的理解，才能够做任何有用的事情。

为了跟得上竞争的需求，以及能够满足更多的项目需求，PHP 和 MySQL 也必须发展。和 2001 年相比，PHP 现在是一种更加复杂和强大的语言，MySQL 是一种更加复杂和强大的数据库。今天，学习 PHP 和 MySQL 将会打开更多的大门，而这些大门在 2001 年的时候对 PHP 和 MySQL 的专家也是关闭的。

这是好消息。坏消息是，切黄油的刀总是比瑞士军刀更容易搞清楚如何使用。同样的道理，这些耀眼的新功能和改进必然使得 PHP 和 MySQL 对于初学者来说变得更加难以学习了。

1 万小时的练习才能成为专家

这种说法背后的科学道理是值得怀疑的，但是其感性认识是正确的。编程是一种技能，并且它肯定很难掌握。不要指望一夜就能变得技艺纯熟。在学习完本书后，你将会很好地理解 PHP，但是，不管你达到了什么层级，总是有很多知识要学习。

有一种说法是，在编程领域，一点点知识就可以用很长的时间。只需要掌握几样工具，就能够做如此之多的事情，你可能会对此感到惊讶。你会发现，在学习了非常基础的知识之后，你就可以实现几乎想要做的任何事情。即便你只是知道超出本书之外的一点点编程概念，不能做的事情非常少。较为高级的概念都是关于让你的代码变得更高效、更快速、更容易编写，并且能够更简单地在其上进行构建的。

稳扎稳打，莫贪走捷径

我再三对那些错过了讲课的学生强调这一点。一个编程概念总是构建在另一个概念之上。大多数时候，你需要学习前面的概念，然后才能继续学习后面的概念。如果你试图走得太快，很容易混淆所学的概念，并且使得自己的学习过程变得困难。

没有多少编程概念是独立存在的，如果你遇到困难，通常是没有完全理解前面的概念所导致的。不要害怕重新复习那些你认为之前已经知道的知识。当你遇到困难的时候，全面地掌握知识通常比费力地学新知识更有效。

协和谬误

20 世纪 70 年代末，英国和法国政府持续给协和飞机投入资金，即便它已经浪费了大量的金钱。他们的理由是，已经在该项目上花了这么多的钱，如果放弃该项目，那么之前投入的资金都会付之东流。当然，他们最终失去了更多，因为他们后来还投入了资金。如果他们早点停下来，从长远来讲，会节省很多的钱。这种现象通常称为“协和谬误”。很多时候，砍掉你失去的东西比继续在失败的项目上投入要好一些。

当你在某件事情上花了很多的时间但它还是不能工作的时候，就到了这种时候了。当发生这种情况时，退后一步并且尝试以一种不同的方式来解决。使用一些你所掌握的替代性的工具。解决方案可能不够优雅，但是一旦能让其工作，你就扭转了败局。

不要害怕放弃任何事情并重新开始。当你开始编程的时候，最终会编写很多的代码，尝试让它们和之

前已经完成的工作协作，并且逐渐构建出了一个“怪物”。你并不是真的理解代码要做什么工作。代码将变得无法工作，并且你会垂头丧气。即便做出最少的修改，也是很困难的工作，因为这很可能会破坏其他的事情。

当发生这种情况的时候，不要害怕从头开始。我已经记不清有多少次这样的情况，当我已经完成了项目的一部分时，又不得不从头开始这个项目。通常在几个小时之内，你会遇到困难并面临相同的处境，但是，你有了更加整齐的代码以及对问题有更好的理解。然而，我强烈建议你使用原来的代码作为参照，而不是删除它。

每个人都是从编写难看的代码开始的。让任何程序员去看他们刚刚入门时编写的代码，他们都会感到尴尬，即便他们是几个月前才刚开始从事编程工作的。

你不是在学习 PHP

是的，没错。本书关注于 PHP 和 MySQL，但是，不要陷入认为你在学习 PHP 的思路中。好吧，你是在学习 PHP，但是，我只不过是使用 PHP 教你编程。

当你学习开车的时候，你不是在学习如何开福特车。你在学习开车的概念，并且可以把这些概念应用到你所驾驶的任何汽车上，即便有一些操控的位置是不同的。

我们在这里学习的概念，将适用于你未来想要学习的几乎任何语言。当然，这里有一些差异，但是，底层的概念是相同的。

一旦你能够用一种语言熟练地编程，只需要几天的时间，你就会对另一种语言的理解达到一个合理标准。因此，不要抱着“我在学习 PHP”的思想阅读本书，相反，要这样想：“我是在学习编程。”

更重要的是，记住概念而不是语法。你总是能够查找正确的语法，但是，理解底层的概念更难。这会把我们带到下一个重点……

把括号和分号放到正确的位置是容易的事情

在学习编程之初，你总是会吧方括号、花括号、分号、点号和相当多的其他内容放错地方。你可能会忘记使用某个单个的字符，从而导致整个程序无法工作。

刚开始的时候，这可能特别令人沮丧。但是，一旦你掌握了它，很快会意识到让语法正确是很容易的事情。这很容易，因为它很严格。语法要么正确，要么不正确。代码要么有效，要么无效。

困难的部分实际上是编写逻辑，将一个问题分解为最小的部分，以便可以将其解释给计算机。计算机将会快速地告诉你语法是否正确，但是它无法告诉你是否给出了正确的解决问题的指令。

规划并不能做成任何事情

规划并不能做成任何事情。

——Karl Pilkington

如果你已经读过有关编程的图书，你可能已经听说过，需要花很多的时间来设计你的代码，因此在编写任何一行代码之前，应该仔细地规划程序的逻辑以及它将会如何工作。你还将会遇到一些教授一种叫作“需求工程”的开发方法学的书和文章，可视化地表示代码的某种图，以及在编写代码之前如何规划代码的各种技巧。

我现在要说一些让大多数程序员感到痛苦的事情——完全忽视这些建议，并且直接投入到代码的编写中。

当我在讲课时这么说的时，我的学生松了一口气。他们来这里学习编写代码的，而学习编写代码

最好的方法就是开始编写代码。

强调先规划设计的建议的基本问题是，它忘记了一个显著的事实：要设计软件，需要知道什么工具可用以及它们所能解决的问题。否则，如果你不知道哪些工具可用的话，所提出的设计将毫无意义。

假设你对于盖房子一无所知。你不知道如何使用锤子、锯子；不知道要支撑屋顶的话，一个房梁需要有多么的坚固；不知道地基需要挖多深；不知道浴室里如何安装水泵；甚至不知道房屋的每一个部分适合用什么样的材料来构建，等等。

只要你愿意，可以在设计上花费尽可能长的时间，并且尽可能仔细地规划事情，但是，除非你知道你的工具能够做什么以及它们的局限性，你最终得到的是一个无法完全利用工具的设计，或者说使用可用的工具/材料根本不可能实现这个设计。如果不知道3层楼的房子需要6米深的地基的话，你根本无法设计一栋3层的房子。

同样，如果你不知道如何编程的话，就无法设计计算机程序。为了说明我的观点，下面来讲讲来自“TED Talk”节目的一个故事，这是Ernesto Strolli讲的一个故事，故事的名字是“想要帮助别人吗？那就别说话并聆听”。

曾经有一个项目是意大利人决定教赞比亚人如何种植粮食的。我们带着意大利的种子到达了赞比亚南部一个非常大的山谷，这个山谷一直向下延伸到赞比西河。让我们感到惊讶的是，当地的人们位于这样一个土壤肥沃的山谷，却没有种植任何农作物。但是，我们并没有问他们为什么没有种植任何作物，而只是说：“好在我们来到了这里。现在就是拯救赞比亚人于饥饿的历史性时刻了。”

当然，一切作物在非洲长得很好，并且我们种植了规模壮观的土豆。在赞比亚，土豆长得甚至比在意大利还要大。我们告诉赞比亚人，看看吧，种植农作物是多么容易啊。当土豆长势喜人，即将成熟时，到了晚上，200多只河马从河里出来并且吃光了所有的东西。我们对赞比亚人说：“天啊，这里有河马。”赞比亚人说：“是的。这就是我们没有在这里种植农作物的原因”。

Ernesto的团队完全知道自己要做什么。他们仔细地规划了一切，并且设法得到了想要的结果。然而，由于他们没有预见到的事情，所有的规划和设计都浪费了。程序员不会遇到河马，但是，有很多的障碍是他们所无法预料的，并且不可避免地会遇到这些障碍。任何时候，一旦相当于200只河马的事情发生了并且“吃掉了”你的代码，你花在设计上的所有时间也都白白浪费了。你必须放弃原来的设计并且重新开始。

在本书中，我会就你可能遇到的各种“河马”给出警示。但是，自行测试更好。你可以“做中学”，投入进去，编写一些代码。第一次的时候，这些代码几乎肯定无法工作，但是你会从这个过程中学到一些东西。用不同的方式再次尝试，有助于你提出一些能够让代码工作的解决方案。

只有意识到可能遇到的问题以及可用工具的局限性，你才有办法设计出程序来。

好吧，设计并非一无是处

对于专业程序员来说，在构建代码之前先进行设计，这是至关重要的。然而，专业人士要编写的代码，可能需要数年甚至十几年才能编写出来。他们编写的代码需要以这种方式来编写，即代码要具有可扩展性，并且让其他人容易理解。

在本书中，我将带你思考代码的架构以及如何编写可复用和可扩展的代码。但是，你在这里编写的代码并不是要用于真正的项目中并需要维护数年再交付使用。你是在这里学习编程。那就勇往直前并找出所有那些“河马”。你将从错误中学到很多知识，这比从那些立即就能工作的代码中学到的更多。

花费在代码设计上的时间，应该和你的编程能力成正比。如果你只是刚刚开始，只要你对于想要程序做什么有一个大致的理解，就可以投身进去并开始编写代码，直到它确实做你想要它做的事情。你可能会遇到困难，并尝试一种不同的方式，并且不会感觉到你在做错事，因为这和你花在设计上的时间是对应的。我前面说的“协和谬误”在这里也是适用的。

对于前几章，至少直接投入并尝试。运行你的代码，看看它是否能够工作。在我给出解决方案之前，

请尝试解决一些我所设定的问题。通过找出自己的解决方案而不是闭着眼睛录入我给你的代码，你将会学到更多。

随着知识的增长，你将会对于哪些工具可用以及分解问题所需的方法都会有一个牢固的理解。一旦达到了这个级别，你就可以开始在编写代码之前更加详细地进行规划了。

本书所使用的体例

你将会注意到，本书使用了特定的排版样式，以强调不同类型的信息。注意如下的一些内容。

代码实例

本书中的代码用等宽字体显示，如下所示：

```
<h1>A Perfect Summer's Day</h1>
<p>It was a lovely day for a walk in the park.
The birds were singing and the kids were all back at
  school.</p>
```

如果这段代码可以在本书配套的示例代码文件中找到，将会在程序列表的顶部给出该示例的名称，如清单 0-1 所示：

清单 0-1 Layout

```
.footer {
  background-color: #CCC;
  border-top: 1px solid #333;
}
```

有些代码行应该是在一行中输入，但是，由于页面宽度有限我们必须折行。➡表示为了版式的目的而进行换行，实际上，该换行应该被忽略：

```
URL.open("http://www.sitepoint.com/responsive-web-design-real
  ➡ -user-testing/?responsive1");
```

提示、注意和警告



提示将给出一些有用的知识点。



注意表示和讨论的主题相关，但不是至关重要的信息。可以将它们当作额外的相关信息。



请注意这些重要的知识点。



警告、强调任何你可能会犯错的陷阱。

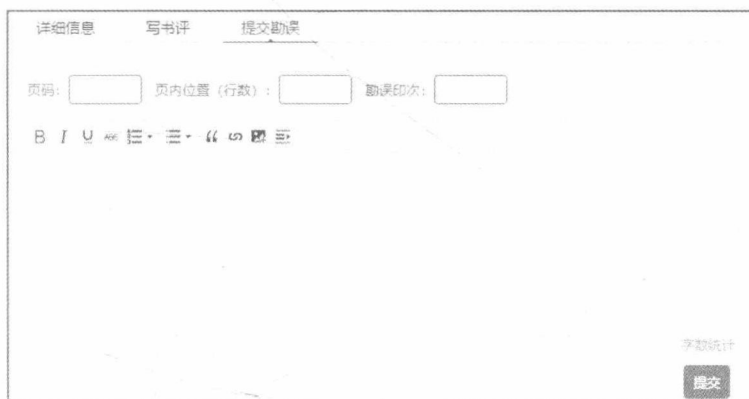
资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，单击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，将赠予您异步社区的 100 积分（积分可用于在异步社区兑换优惠券、样书或奖品）。



扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书

名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

目 录

第 1 章 安装	1	3.7 一点警告	49
1.1 属于自己的 Web 服务器	1	3.8 查看存储的数据	50
1.1.1 服务器安装方法 1: 手动安装 所有的软件	2	3.9 修改存储的数据	52
1.1.2 服务器安装方法 2: 预打包安装	2	3.10 删除存储的数据	53
1.1.3 服务器安装方法 3: 虚拟服务器	3	3.11 让 PHP 进行输入	53
1.2 你自己的虚拟机	4	第 4 章 在 Web 上发布 MySQL 数据	55
1.2.1 在 Windows 上安装	4	4.1 蓝图	55
1.2.2 在 macOS 上安装	4	4.2 创建 MySQL 用户账户	56
1.2.3 在 Linux 上安装	4	4.3 使用 PHP 连接 MySQL	58
1.3 启动 Vagrant	4	4.4 面向对象编程一瞥	61
第 2 章 PHP 简介	8	4.5 用 PHP 发送 SQL 查询	64
2.1 基本的语法和语句	9	4.6 处理 SELECT 结果集	67
2.2 变量、操作符和注释	11	4.7 提前考虑	71
2.2.1 变量	11	4.8 将数据插入数据库	78
2.2.2 操作符	11	4.9 从数据库删除数据	84
2.2.3 注释	12	4.10 完成任务	89
2.3 控制结构	12	第 5 章 关系数据库设计	90
2.3.1 if 语句	12	5.1 该署名时署名	90
2.3.2 循环	16	5.2 首要原则: 保持实体分离	92
2.4 数组	21	5.3 查询多个表	94
2.5 用户交互和表单	25	5.4 简单关系	98
2.6 在表单中传递变量	29	5.5 多对多关系	99
2.7 弥补缝隙	31	5.6 一对多和多对一	101
2.7.1 使用 PHP 模板	31	第 6 章 PHP 结构化编程	102
2.7.2 安全性问题	32	6.1 包含文件	102
2.7.3 多个模板, 一个控制器	34	6.1.1 包含 HTML 内容	102
2.8 接触数据库	36	6.1.2 包含 PHP 代码	103
第 3 章 MySQL 简介	37	6.1.3 包含的类型	105
3.1 数据库简介	37	6.2 定制函数和函数库	106
3.2 使用 MySQL Workbench 运行 SQL 查询	38	6.3 将代码分解为可复用的函数	112
3.3 创建数据库	41	6.3.1 使用函数来替代查询	115
3.4 SQL 语言	42	6.3.2 更新笑话	117
3.5 创建表	44	6.3.3 在 Web 站点上编辑笑话	117
3.6 将数据插入表中	46	6.3.4 删除函数	119
		6.3.5 选择函数	120
		6.4 最佳方法	121

第 7 章 改进插入和更新函数	122	9.15 Composer 简介	206
7.1 改进更新笑话的函数	122	9.16 REST 简介	206
7.1.1 去除末尾的逗号	124	9.17 使用接口增强依赖性结构	211
7.1.2 改进插入笑话的函数	125	9.18 你自己的框架	213
7.2 处理日期	127	第 10 章 允许用户注册账户	214
7.3 自己制作工具	132	10.1 验证 E-mail 地址	219
7.4 通用型函数	133	10.2 防止同一个人注册两次	220
7.5 使用这些函数	137	10.3 安全地存储密码	222
7.6 重复的代码是敌人	139	10.4 注册完成	224
7.7 进一步打磨	143	第 11 章 cookie、session 和访问控制	227
7.8 继续前进	145	11.1 cookie	227
第 8 章 对象和类	146	11.2 PHP 会话	230
8.1 类	146	11.3 访问控制	232
8.2 公有和私有	149	11.3.1 登录	233
8.3 对象	149	11.3.2 受保护的页面	237
8.4 类变量	150	11.3.3 接口和返回类型	239
8.5 构造方法	154	11.3.4 使用验证类	241
8.6 类型提示	155	11.3.5 登录错误消息	241
8.7 私有变量	157	11.4 创建一个登录表单	242
8.8 使用 DatabaseTable 类	158	11.5 退出	245
8.9 更新控制器以使用类	161	11.5.1 给登录用户增加添加笑话的 功能	246
8.10 DRY	163	11.5.2 用户许可	248
8.11 创建一个控制器类	164	11.5.3 任务是否完成了	249
8.12 单点入口	167	11.6 天高任鸟飞	251
8.13 保持 DRY 原则	171	第 12 章 MySQL 管理	252
8.14 模板变量	174	12.1 备份 MySQL 数据库	252
8.15 小心 extract	176	12.1.1 使用 MySQL Workbench 进行 数据库备份	253
第 9 章 创建一个可扩展的框架	178	12.1.2 使用 mysqldump 进行数据库 备份	254
9.1 搜索引擎	179	12.1.3 使用二进制日志的增量备份	255
9.2 让内容更加通用	180	12.2 MySQL 访问控制技巧	256
9.3 提前考虑: 用户注册	181	12.2.1 主机名问题	257
9.4 依赖性	184	12.2.2 锁在外面了	258
9.5 URL 重写	188	12.3 索引	259
9.6 整理	190	12.4 外键	263
9.7 在不同站点上复用代码	194	12.5 安全比说抱歉好	264
9.8 通用的, 还是特定于项目的?	194	第 13 章 关系	265
9.9 自动加载和命名空间	199	13.1 对象关系映射器	267
9.10 区分大小写	200	13.1.1 公有属性	268
9.11 实现自动加载程序	201		
9.12 重定向	201		
9.13 命名空间	203		
9.14 用 PSR-4 自动加载	204		

13.1.2 实体类中的方法	268	13.8.8 在数据库中存储按位许可	308
13.1.3 使用来自 DatabaseTable 类的 实体类	270	13.8.9 连接表或按位计算	309
13.1.4 笑话对象	274	13.8.10 整理	309
13.1.5 使用 Joke 类	276	13.8.11 编辑其他人的笑话	310
13.1.6 引用	276	13.9 大功告成	312
13.1.7 简化列表控制器动作	277	第 14 章 用正则表达式进行内容格式化	313
13.2 整理	279	14.1 正则表达式	313
13.3 笑话分类	280	14.2 用正则表达式进行字符串替换	315
13.4 指定笑话的分类	284	14.2.1 强调文本	315
13.5 按照分类显示笑话	293	14.2.2 段落	318
13.6 编辑笑话	296	14.2.3 超链接	320
13.7 用户角色	298	14.3 综合应用	321
13.8 创建一个表单以分配许可	302	14.4 排序、限定和偏移	324
13.8.1 作者列表	302	14.4.1 排序	324
13.8.2 编辑作者权限	303	14.4.2 用 LIMIT 和 OFFSET 分页	327
13.8.3 设置许可	304	14.4.3 分类中的分页	331
13.8.4 一种不同的方法	305	14.5 达到专业水准	334
13.8.5 快速了解二进制	306	附录 A 使用示例代码	335
13.8.6 位计算能力	306	附录 B Linux 故障排除	336
13.8.7 回到 PHP	307		

第 1 章 安 装

在本书中，我们将帮助你跨出超越静态页面构建的第一步。静态页面是使用 HTML、CSS 和 JavaScript 这样的纯客户端技术构建的。我们将一起探索数据库驱动的 Web 站点的世界，看看令人眼花缭乱的动态工具、概念以及它们所带来的各种可能。

在开始构建第一个动态 Web 站点之前，你必须收集完成这项工作所需的工具。这就像是在烘焙蛋糕之前，你需要准备好制作蛋糕的配料，然后才能按照菜谱来制作。在本章中，我们将介绍如何下载和安装所必需的软件包。

如果你过去经常使用 HTML 和 CSS，甚至是用更加智能化的 JavaScript 来构建 Web 站点，你可能很熟悉将文件上传到某个位置以组成站点的方式。这可能是你已经付费的虚拟主机服务，由你的互联网服务提供商（Internet Service Provider, ISP）提供的 Web 空间，也可能是由你所在的公司 IT 部门搭建的 Web 服务器。在任何情况下，一旦你将自己的文件复制到这些目的地中的任何一个，当 Internet Explorer、Google Chrome、Safari 或 Firefox 这样的 Web 浏览器请求这些文件的时候，常用的 Web 服务器的软件程序就能够找到并提供这些文件的副本。你可能听说过的 Web 服务器软件程序，包括 Apache HTTP Server（Apache）、NGINX 和 Internet Information Services（IIS）。

PHP 是一种服务器端脚本编程语言。你可以将 PHP 看作 Web 服务器的一个插件，它使得 Web 服务器能够做更多的工作，而不只是准确地发送 Web 浏览器所请求的文件副本。安装了 PHP 之后，Web 服务器将能够运行小程序（叫作 PHP 脚本），执行诸如此类的任务：从数据库提取最新的信息，用这些信息生成一个实时的 Web 页面，然后将其发送给请求该页面的浏览器。本书的大部分内容将集中介绍如何编写 PHP 脚本来做这样的事情。PHP 可以完全免费地下载和使用。

要让 PHP 脚本从数据库获取信息，必须首先有一个数据库。这就是 MySQL 的用武之地。MySQL 是一种关系数据库管理系统（Relational DataBase Management System, RDBMS）。稍后，我们将介绍 MySQL 的具体角色以及它是如何工作的。简言之，它是一种软件程序，能够高效地组织和管理众多的信息片段，同时记录这些信息片段之间是如何彼此关联的。MySQL 还使得诸如 PHP 这样的服务器端脚本编程语言能够非常容易地访问那些信息，并且和 PHP 一样，MySQL 也是完全免费使用的。

本章的目标是建立配备了 PHP 和 MySQL 的一个 Web 服务器。我将一步一步地指导你在最新的 Windows 和 Mac OS X 系统上工作，因此，不管你喜欢使用什么样的计算机，这里都有你所需的内容。

1.1 属于自己的 Web 服务器

如果你当前的虚拟主机的 Web 服务器已经安装了 PHP 和 MySQL，那么，你很幸运。大多数虚拟主机确实会这么做，这也是 PHP 和 MySQL 如此流行的原因之一。如果是这样，那么，好消息是，你可以发布自己的第一个数据库驱动 Web 站点，而不必购买任何支持相应技术的虚拟主机服务。

在开发静态 Web 站点时，你可以直接从硬盘将 HTML 文件加载到自己的浏览器中，预览它们的显示效果。当你这么做的时候，没有 Web 服务器软件也没问题，因为 Web 浏览器自身就能够阅读和理解 HTML 代码。

然而，当使用 PHP 和 MySQL 来构建动态 Web 站点时，Web 浏览器则需要一些帮助。虽然 Web 浏览器无法理解 PHP 脚本，然而，PHP 脚本包含了针对能够理解 PHP 的 Web 服务器的指令，通过执行这些指令可以生成浏览器能够理解的 HTML 代码。

即便你已经有了一个支持 PHP 的 Web 主机，你仍然想要能够自己运行 PHP 脚本，而不需要使用其他人的服务器。为此，你需要安装自己的 Web 服务器。“服务器”这个术语可能使你想到一个较大的带有空

调的房间，其中的机架上装满了很大的计算机。但是不要担心，你不需要任何全新的硬件。笔记本电脑或台式机就能很好地工作了。

要在自己的 Web 主机上运行 PHP 脚本，你需要在编辑器中编写它们，打开 FTP 或 SSH 客户端并且将它们上传到服务器上。然后，才可以导航到你所创建的文件 URI，以便在浏览器中看到结果。如果你犯了一个错误并且脚本有错的话，你将需要修改代码，回到 FTP 程序，再次上传文件，然后重新加载页面。这个过程很烦琐，并且占用你本可以用来编写代码的宝贵时间。通过在自己的 PC 上运行服务器，你将能够在编辑器中保存文件，并且通过直接刷新页面在浏览器中查看修改，而不需要上传文件。即便你已经有了一个完美的 Web 主机，在 PC 上运行服务器也真的很节省时间，并且也是其最大的一个优点（尽管还不是唯一的优点）。

那么，该如何在你的 PC 上运行 Web 服务器呢？有 3 种方法可以做到这一点，每一种方法都有其自己的优点和缺点。

1.1.1 服务器安装方法 1：手动安装所有的软件

Apache 是一个 Web 服务器，像大多数的软件一样，它带有一个安装程序，使你可以很容易地将其安装到自己的 PC 上。不需要费太多的力气，你就可以用它提供 Web 页面了。然而，有成百的配置选项，并且，除非你知道自己在做什么，否则要让其可供开发 PHP Web 站点使用还是颇费时间并且容易令人混淆的。

要运行 PHP 脚本，只有一个 Web 服务器是不够的。要进行手动安装，你还需要安装 PHP（它并没有一个安装程序）并配置它。和 Apache 一样，PHP 也有很多的选项，并且它默认地针对你想要运行一个真实的 Web 站点而安装。对于开发代码来说，这很糟糕，因为不会显示出错误。如果你犯了一个错误，将会得到一个空白的页面，而不会有任何指示表明哪里出错了。即便只是错了一个字符（例如，漏掉了一个花括号或者分号），也会得到一个空白页面，不会提示是什么导致了这个问题。要解决这个问题，你需要手动配置 PHP 安装，并且将设置调整为显示错误消息且支持其他的工具，从而使得开发成为更为愉快的任务。

还需要将 Apache 配置为能够和 PHP 通信，以便当某人连接到服务器并请求一个扩展名为 .php 的文件时，该文件首先发送给 PHP 以进行处理。

为了学习本书内容，你还需要用到 MySQL，这意味着也要进行手动安装和配置。

Apache、MySQL 和 PHP 每一个都有数十项配置，并且除非你知道自己在做什么，这些配置是很难设置的。即便你是一位专家，也至少要花 1 个小时才能让一切都能工作。

手动安装需要大量的知识和研究工作，这些已经超出了本书的讨论范围。尽管能够配置服务器是一项有用的技能，但这对于帮助你学习如何使用 PHP 编程并没有帮助，而当你阅读本书时，真正感兴趣的正是如何使用 PHP 编程。

这种选择不仅适合于胆怯的新手，对于有经验的专业人士来说也是适合的，因为很容易漏掉一些重要的设置。好在我们并不需要关心所有这些软件的单个安装和配置工作。

1.1.2 服务器安装方法 2：预打包安装

多年以来，开发者组织认识到了手动安装的问题所在，并且他们编译了预打包安装程序，这是一个单个的安装程序，它可以安装 PHP、Apache、MySQL 和其他的相关软件，所有的预配置都针对你这样的开发者做了适当的配置。预打包的安装包的一些示例如 XAMPP（X、Apache、MySQL、PHP、Perl）、WAMP（Windows、Apache、MySQL、PHP）和 LAMP（Linux、Apache、MySQL、PHP）。

显然，这比手动安装每一个软件要容易多了，并且不需要学习如何配置服务器。这比手动安装要快很多，也要容易很多，尽管使用这种方法的时候，你仍然可能会遇到如下的问题。

（1）你的 Web 托管主机可能运行 Linux，但是，你的 PC 可能并不是这样的。尽管 Apache、MySQL 和 PHP 能够在 Windows、Linux 或 MacOS 下工作，但是，操作系统的工作方式有很大的差异。在 Windows 上，文件名是不区分大小写的，这意味着，FILE.PHP 与 file.php 和 fIe.pHp 是相同的。在你的 Web 托管主机上，几乎肯定不是这种情况。这经常引发这样的问题——一段脚本在 Windows 开发服务器上工作得很好，但是在上传以后却无效了，因为在代码中引用文件的大小写方式不对。

（2）Apache 和 MySQL 都是服务器，并且它们在后台运行。即便是当你并不开发软件的时候，它们仍

将会运行，耗尽你的计算机的 RAM 和处理能力。

(3) 预打包的软件总是会略微有些过时。尽管对于开发用计算机来说，安全修复并不是首要的问题（你不应该允许人们通过 Web 访问该机器），开发者坚持使用最新版本的软件总是很有用的，这样能够检查 Web 托管主机上的软件更新时可能遇到的问题。如果你的 Web 托管主机使用的 PHP 版本，比你的开发服务器上的 PHP 版本还要新，这可能会由于某些功能已经修改或删除了而导致问题。最后，开发者很喜欢在新的功能发布时就使用它。如果你没有使用最新的版本的话，就不能做到这一点。

尽管预打包安装要比手动安装更好，这些问题并不会使得它们成为理想的解决方案。好在，还有更好的办法。

1.1.3 服务器安装方法 3：虚拟服务器

安装并运行服务器的第 3 种方法是使用虚拟服务器。虚拟服务器的行为就像是在另一台不同的计算机上的 Web 服务器。你可以在这台计算机上运行任意的操作系统，并且可以从你的 PC 连接到它，就好像它真的在世界上的某个地方一样。

诸如 VMWare 和 VirtualBox 这样的虚拟化软件很常见。作为一名 Web 开发人员，你可能熟悉诸如 modern.ie 这样的工具，这是 Microsoft 提供的一种有用的服务，可以下载运行在 Windows、Microsoft Edge 和 Microsoft Edge 的各种版本之上的虚拟机。如果你想要看看自己的 Web 站点在 Windows XP 上的 Internet Explorer 8 中的样子，可以下载相关的虚拟机并且在你的 Window10、macOS 或 Linux 桌面的计算机上一个窗口中运行它，而不需要真正地在已有的 Window 10、macOS 或 Linux 之中再安装并运行 Windows XP 和 Internet Explorer 8。

像 VirtualBox 这样的软件，允许你在一个操作系统之中运行另一个操作系统，如图 1-1 所示。要测试 Internet Explorer 8，你可以在一个虚拟机中运行 Windows 7。然而，我们需要运行 PHP 脚本，这就允许我们做一些更炫酷的事情——我们可以运行一个带有 PHP、Apache 和 MySQL 的 Linux Web 服务器，而这个服务器是安装有 Windows 或 macOS 的 PC。

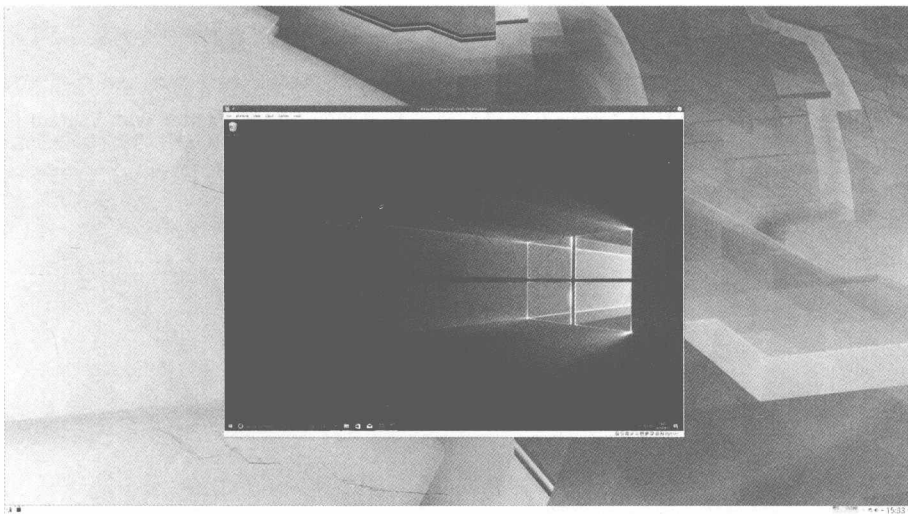


图 1-1 在 Arch Linux 中运行的 Windows 10

这就允许你运行和 Web 托管主机上所运行的完全相同版本的 PHP、MySQL 和 Apache，而且让它们运行于完全相同的操作系统之上，这就防止了由于所采用的操作系统的版本差异而导致的任何问题。

最大的一个优点是，你可以下载预配置的虚拟机，例如，微软所提供的 Windows XP 和 Internet Explorer 8 虚拟机，或者将 PHP、Apache 和 MySQL 安装和配置为一起工作的一个虚拟机。这就像是预配置的软件包在 Linux 上运行，就好像这是你的网络上的一个真正的 Web 服务器一样。

所有这些做法的缺点是，要运行你的代码，你必须下载一个完整的操作系统。这意味着要进行更多的下载，但是，在 10Mbit/s 的互联网带宽和太字节的硬盘空间的年代，这真的是无须担忧的问题。因为这是