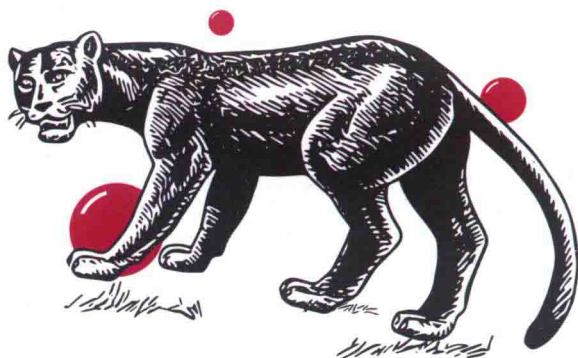




本书源代码下载

- 曾供职360后端系统开发、现今日头条大数据工程师专业奉献
- 使用易于理解的代码示例详解PHP 7的使用和新特性
- 运用MySQL、Redis、ThinkPHP、常用设计模式及流行项目案例提高编程技能



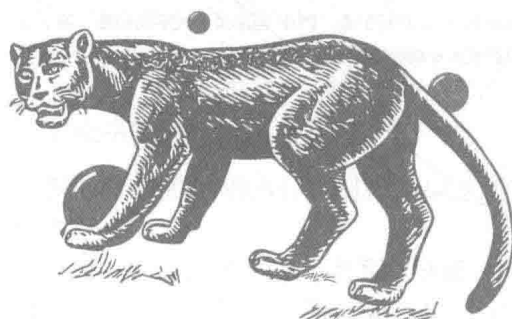
PHP 7

从零基础到项目实战

陈小龙 编著



机械工业出版社
China Machine Press



PHP 7

从零基础到项目实战

陈小龙 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

PHP 7从零基础到项目实战/陈小龙编著. —北京: 机械工业出版社, 2018.9

ISBN 978-7-111-61050-2

I. ①P… II. ①陈… III. ①PHP语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字 (2018) 第226209号

本书根据编者多年 PHP 开发经验精心编写而成。全书从基础到实践, 循序渐进地介绍了 PHP 7 编程的核心知识与技术要点, 主要内容包括: PHP 语法基础、函数、数组、表单、对象、XML 和 JSON、Cookie 和 Session、异常处理、PHP 类、正则表达式、常见的设计模式、PHP 操作图像和文件、MVC 架构思想、ThinkPHP 框架、PHP 操作 Redis 和 MySQL 数据库。此外, 还介绍了基于前端架构打造服务端、自己动手编写 PHP 框架、O2O 网站开发和 App 后台开发以及大型网站架构设计等。

本书内容丰富, 技术先进, 适合 PHP 开发初学者、PHP 程序员、大学生和想了解 PHP 7 新功能的开发人员使用, 也很适合作为培训班、企业内训或大专院校的教学用书。

PHP 7 从零基础到项目实战

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 夏非彼 迟振春

印 刷: 中国电影出版社印刷厂

开 本: 188mm×260mm 1/16

书 号: ISBN 978-7-111-61050-2

责任校对: 王叶

版 次: 2018 年 10 月第 1 版第 1 次印刷

印 张: 24.5

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzit@hzbook.com

版权所有•侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东



前言

2015年6月发布了PHP 7 Alpha 1版本，同年12月3日发布了GA版本，PHP 7的发布对于PHP来说具有里程碑的意义。在性能上，PHP 7的执行效率是原来PHP 5的两倍左右。PHP 7的底层架构改变了存储各种变量的Zval和Zend_String结构体，优化了Zend_Array的HashTable等，这些底层架构的改进大幅度提升了PHP的执行效率。

本书的目的是帮助读者快速学习和掌握PHP 7，从最基础的PHP开发环境的搭建开始，逐渐深入到PHP的相关特性和Redis及MySQL等的使用，特别是介绍了ThinkPHP、常见的设计模式、MVC设计思想等，最后还介绍了基本前端架构打造服务端、App后台开发、框架编程等较流行的开发案例，以帮助读者学以致用，真正开启PHP开发之旅。

本书结构

本书共分24章，各章内容说明如下：

第1章为走进PHP的世界，介绍PHP的运行机制和PHP 7的新特性，学习搭建PHP的开发环境，并编写第一个PHP程序。

第2章介绍PHP的语言基础，包括PHP的数据类型、变量、常量和运算符的知识。

第3章讲解PHP中的流程控制语句，介绍foreach语句在PHP 7和以往版本中的不同。

第4章介绍函数的使用，包括函数参数的传递方式、可变函数和匿名函数等，与PHP 5不同的是，PHP 7中新增了支持参数类型的声明和函数返回值类型的声明。

第5章详细介绍PHP中的字符串，并着重讲解编程中经常会用到的一些字符串处理函数。

第6章讲解PHP数组有关的内容，和字符串一样，数组也是我们在编程中经常用到的。

第7章讲解PHP中的时间与日期函数，包括如何设置和获取时间、如何计算两个日期的时间差等。

第8章介绍表单。PHP作为一种动态语言，经常需要收集前端用户传过来的数据，然后与数据库交互，表单是用户填写数据、发起与数据库交互的第一步。

第9章是类与对象，介绍什么是类及类的使用。学会使用类封装一些方法，具备面向对象编程的思想，是开发大型网站必不可少的基本功。

第10章介绍正则表达式有关的内容。几乎所有的编程语言都支持正则表达式，本章讲解正则表达式的基本内容以及如何在PHP中使用正则表达式。

第11章介绍PHP中的错误异常处理。PHP 7中增加了错误处理，增加了一个Error类。

第12章介绍如何使用PHP处理图像，如获取图像信息、拷贝旋转图像、给图像加水印等。

第13章是目录文件操作。PHP有着强大的目录文件操作函数，可以使用它们创建、修改、读取文件，还可以改变文件的属性等，另外还介绍了与文件上传有关的配置。

第14章详细讲解Cookie和Session，介绍它们的基本概念和设置，通过实际的案例说明它们的工作原理和存储机制。

第15章介绍MySQL数据库的使用。这一章介绍数据库的安装和MySQL的一些基本操作，以及如何使用PHP与MySQL交互。本章在编写的时候摒弃了PHP 5版本中与MySQL连接的MySQL扩展，转而重点介绍PHP如何使用MySQLi和PDO与数据库交互。

第16章介绍Redis的使用，介绍Redis的5种数据类型，并讲解如何使用PHP操作Redis。

第17章是PHP处理XML和JSON，介绍几种创建与读取XML的方式，以及PHP中json_encode()和json_decode()函数的使用。

第 18 章介绍 MVC 思想和国内流行的 ThinkPHP 框架，本章介绍的是 ThinkPHP 的最新版本，和以往的版本有许多不同，读者在阅读时需注意。

第 19 章介绍编程中几种常用的设计模式，包括工厂模式、单例模式、观察者模式和策略模式。

第 20 章是基于前端架构打造服务端，介绍如何使用 API 接口与前端交互，传输消息的加解密，另外还介绍了前端开发中常用的模板 MustacheJS 和 AngularJS。

第 21 章从零开始讲述一个 O2O 网站的开发流程，从需求分析到数据库设计，再到最后的编码实现，以及如何应用支付模块等。

第 22 章介绍当今比较流行的混合式 App 的开发框架，以及如何开发接口程序、如何定义路由等。

第 23 章讲解如何自己动手实现一个 PHP 开发框架，介绍框架的开发思想和流程，以及框架的基本概念和基础结构。自己实现一个框架对于理解框架的运行原理有很好的促进作用。

第 24 章讲述目前大型网站中常用的技术栈，通过这一章的学习，读者可以开阔眼界，对大型复杂的网站系统有更好的了解。

读者对象

- PHP 初学者
- 想了解 PHP 7 新特性的读者
- 想进阶的 PHP 程序员
- 开设相关课程的大专院校学生
- 公司内部培训的学员

技术支持

在编写本书的过程中，笔者一直努力为读者呈现完整的知识体系结构，但由于本人水平和开发经验有限，书中难免存在不足之处，还望广大读者指正。如果对书中的内容有建议或疑惑，可通过笔者的微信公众号、邮箱或书友群与笔者联系，笔者一定会尽量解答。

微信公众号：chenxiaolong19941024

邮箱：314312298@qq.com

书友群：201463512

本书配套的素材文件下载地址（注意区分数字和字母大小写）如下：

<https://github.com/chenxiaolong1994/php7book>

如果下载有问题，请发邮件到电子邮箱 booksaga@126.com 获得帮助，邮件标题为“PHP 7 从零基础到项目实战下载资源”。

致谢

首先，感谢 PHP 之父 Rasmus Lerdorf，是他创建了这个优秀的编程语言，今天我们在互联网上浏览的网页有许多都是使用 PHP 编写的。同时希望越来越多的朋友加入到 PHP 的学习和开发中来，共同将这一优秀的编程语言发扬光大。

其次，感谢王金柱编辑的大力支持，他在本书的编辑和出版过程中付出了很多心血，此书得以与读者见面，离不开他的辛勤工作。

最后，还要感谢家人和朋友的支持，写作本书需要耗费许多时间，使得我不能经常陪伴家人和朋友。

陈小龙

2018 年 5 月于北京

目 录

前言

第 1 章 走进 PHP 的世界..... 1

- 1.1 快速认识 PHP.....2
 - 1.1.1 PHP 语言的优势.....2
 - 1.1.2 PHP 的运行机制和原理.....3
 - 1.1.3 PHP 7 的新特性.....4
- 1.2 搭建 PHP 开发环境.....5
- 1.3 常用 PHP 代码编辑器.....7
- 1.4 编写一个“Hello World” PHP 程序.....7

第 2 章 PHP 语言基础..... 9

- 2.1 PHP 的数据类型.....9
 - 2.1.1 字符串.....9
 - 2.1.2 整型.....10
 - 2.1.3 浮点型.....11
 - 2.1.4 布尔型.....11
 - 2.1.5 数组.....11
 - 2.1.6 对象.....11
 - 2.1.7 NULL 值.....12
- 2.2 运算符.....12
 - 2.2.1 算术运算符.....12
 - 2.2.2 递增递减运算符.....13
 - 2.2.3 比较运算符.....14
 - 2.2.4 逻辑运算符.....15
 - 2.2.5 三元运算符.....15
 - 2.2.6 字符串连接运算符.....16
 - 2.2.7 赋值运算符.....16
 - 2.2.8 位运算符.....17
- 2.3 变量.....18
 - 2.3.1 变量的定义.....18
 - 2.3.2 变量的作用域.....18
- 2.4 常量.....20
 - 2.4.1 常量的声明.....20
 - 2.4.2 预定义常量.....21

第 3 章 流程控制语句..... 23

- 3.1 条件控制语句.....23
 - 3.1.1 if 条件控制语句.....23
 - 3.1.2 switch 分支语句.....24
- 3.2 循环控制语句.....26
 - 3.2.1 while 循环.....26
 - 3.2.2 do while 循环.....26
 - 3.2.3 for 循环.....27
 - 3.2.4 foreach 循环.....28
- 3.3 跳转语句.....30
 - 3.3.1 break 语句.....30
 - 3.3.2 continue 语句.....30
 - 3.3.3 goto 语句.....30
- 3.4 包含语句.....31
 - 3.4.1 include 语句.....31
 - 3.4.2 include_once 语句.....33
 - 3.4.3 require 语句.....33
 - 3.4.4 require_once 语句.....34

第 4 章 函数..... 35

- 4.1 函数的使用.....35
- 4.2 函数的参数.....36
 - 4.2.1 参数传递方式.....36
 - 4.2.2 参数类型声明.....38
 - 4.2.3 可变参数数量.....39
- 4.3 函数返回值.....40
- 4.4 可变函数.....41
- 4.5 内置函数.....42
- 4.6 匿名函数.....42
- 4.7 递归与迭代.....44
 - 4.7.1 递归.....44
 - 4.7.2 迭代.....45

第5章 字符串	46	6.4.2 \$_GET 和 \$_POST 数组	86
5.1 单引号和双引号的区别	46	6.4.3 \$_FILES 数组	87
5.2 字符串连接符	47	6.4.4 \$_SESSION 和 \$_COOKIE 数组	88
5.3 字符串操作	47	6.4.5 \$_REQUEST[] 数组	88
5.3.1 改变字符串大小写	47	第7章 时间与日期	89
5.3.2 查找字符串	48	7.1 设置时区	89
5.3.3 替换字符串	50	7.1.1 在配置文件中设置	89
5.3.4 截取字符串	51	7.1.2 通过 date_default_timezone_set 函数在文件中设置	90
5.3.5 去除字符串首尾空格和特殊 字符	51	7.2 获取当前时间	90
5.3.6 计算字符串的长度	52	7.3 常用时间处理方法	93
5.3.7 转义和还原字符串	53	7.3.1 格式化时间显示	93
5.3.8 重复一个字符串	53	7.3.2 计算两个日期间的时间差	95
5.3.9 随机打乱字符串	54	7.3.3 从字符串中解析日期时间	96
5.3.10 分割字符串	54	7.3.4 日期的加减运算	96
第6章 数组	56	7.4 验证日期	97
6.1 使用数组	56	第8章 表单	98
6.1.1 数组类型	56	8.1 表单的种类	98
6.1.2 创建数组	57	8.1.1 文本域及其类型	98
6.2 二维数组和多维数组	59	8.1.2 其他表单类型	102
6.2.1 二维数组	59	8.2 get 和 post 方法	103
6.2.2 多维数组	59	8.2.1 获取表单值	104
6.3 数组操作	60	8.2.2 处理上传文件	105
6.3.1 检查数组中是否存在某个值	60	第9章 类与对象	107
6.3.2 数组转换为字符串	61	9.1 什么是类	107
6.3.3 计算数组中的单元数目	61	9.1.1 声明一个类	108
6.3.4 数组当前单元和数组指针	61	9.1.2 实例化一个类	108
6.3.5 数组中的键名和值	62	9.1.3 访问类中成员	109
6.3.6 填补数组	65	9.1.4 静态属性和静态方法	110
6.3.7 从数组中随机取出一个或 多个单元	69	9.1.5 构造方法和析构方法	111
6.3.8 数组排序与打乱数组	69	9.2 封装和继承特性	113
6.3.9 遍历数组	72	9.2.1 封装特性	113
6.3.10 数组的拆分与合并	76	9.2.2 继承特性	114
6.3.11 增加 / 删除数组中的元素	78	9.2.3 通过继承实现多态	115
6.3.12 其他常用数组函数	79	9.3 魔术方法	116
6.4 系统预定义数组	84	9.3.1 __set() 和 __get() 方法	116
6.4.1 \$_SERVER	84		

9.3.2	__isset()和__unset()方法	117	12.1.3	imagesx 取得图像的宽度	154
9.3.3	__call()和__toString()方法	119	12.1.4	imagesy 取得图像的高度	154
9.4	自动加载	120	12.2	图像绘制	155
9.4.1	__autoload() 方法	120	12.2.1	创建画布	155
9.4.2	spl_autoload_register() 函数	121	12.2.2	定义颜色	156
9.5	抽象类和接口	122	12.2.3	绘制图形	157
9.5.1	抽象类	122	12.2.4	绘制文字	161
9.5.2	接口	123	12.3	图片处理	163
9.6	类中的关键字	125	12.3.1	复制图像	164
9.6.1	final 关键字	125	12.3.2	旋转图像	164
9.6.2	clone 关键字	126	12.3.3	图像水印	165
9.6.3	instanceof 关键字	127	12.4	图像验证码	166
9.6.4	“==” 和 “===”	128	第 13 章	目录文件操作	168
第 10 章	正则表达式	129	13.1	目录	168
10.1	正则表达式的用途	129	13.1.1	判断文件类型	168
10.2	正则表达式的语法	130	13.1.2	创建和删除目录	169
10.2.1	正则表达式中的元素	130	13.1.3	打开读取和关闭目录	170
10.2.2	替换和子表达式	134	13.1.4	获得路径中目录部分	172
10.2.3	反向引用	135	13.1.5	目录磁盘空间	172
10.3	在 PHP 中使用正则表达式	135	13.2	文件操作	173
10.3.1	匹配与查找	135	13.2.1	打开文件	173
10.3.2	搜索与替换	139	13.2.2	读取文件	174
10.3.3	分割与转义	141	13.2.3	获得文件属性	175
第 11 章	错误异常处理	144	13.2.4	复制/删除/移动/重命名 文件	178
11.1	异常处理	144	13.3	文件指针	179
11.1.1	异常类	144	13.4	文件上传	181
11.1.2	创建自己的异常类	146	13.4.1	上传文件配置	181
11.2	错误有关配置	147	13.4.2	上传文件示例	182
11.2.1	错误级别配置	147	第 14 章	Cookie 与 Session	184
11.2.2	记录错误	148	14.1	详解 Cookie	184
11.2.3	自定义错误处理函数	149	14.1.1	Cookie 的基本概念和设置	184
11.3	PHP 7 中的错误处理	151	14.1.2	Cookie 的应用和存储机制	187
第 12 章	图像处理	152	14.2	Session 详解	188
12.1	获取图像信息	152	14.2.1	Session 的基本概念和设置	188
12.1.1	getimagesize 取得图像大小	152	14.2.2	Session 的工作原理和存储 机制	189
12.1.2	getimagesizefromstring 从字符串 中获取图像尺寸信息	153	14.2.3	使用 Redis 存储 Session	190

第 15 章 MySQL 数据库的使用	194	第 17 章 PHP 处理 XML 和 JSON	252
15.1 MySQL 的安装	194	17.1 生成 XML	252
15.2 操作 MySQL 数据库	197	17.1.1 由字符串或数组遍历 生成 XML	252
15.2.1 创建数据库	197	17.1.2 通过 DOM 生成 XML	254
15.2.2 显示数据库	197	17.1.3 通过 PHP SimpleXML 生成 XML	255
15.2.3 选择数据库	197	17.2 解析 XML	256
15.2.4 删除数据库	198	17.2.1 通过 DOM 解析 XML	256
15.3 MySQL 数据类型	198	17.2.2 通过 PHP SimpleXML 解析 XML	258
15.3.1 数值类型	198	17.3 JSON 的使用	259
15.3.2 日期和时间类型	199	17.3.1 json_encode	259
15.3.3 字符串类型	199	17.3.2 json_decode	260
15.4 操作 MySQL 数据表	200	第 18 章 MVC 与 ThinkPHP 框架	261
15.4.1 创建数据表	200	18.1 PHP MVC 概述	261
15.4.2 查看数据表结构	201	18.2 常用的 PHP 框架	262
15.4.3 更改数据表结构	202	18.3 ThinkPHP 的使用	263
15.4.4 删除数据表	204	18.3.1 开始开发	263
15.5 操作 MySQL 数据	205	18.3.2 入口文件与路由	266
15.5.1 插入数据	205	18.4 ThinkPHP 控制器	267
15.5.2 更新数据	205	18.4.1 创建控制器	267
15.5.3 删除数据	206	18.4.2 跳转和重定向	269
15.5.4 查询数据	206	18.5 使用数据库	270
15.6 MySQL 图形化管理工具	210	18.5.1 连接数据库	270
15.7 PHP 操作 MySQL 数据库	212	18.5.2 查询构造器	272
15.7.1 MySQLi 连接操作数据库	212	18.5.3 增加/删除/更新数据	277
15.7.2 PDO 连接操作数据库	215	18.6 模型	279
第 16 章 PHP 与 Redis 数据库	219	18.6.1 模型定义	279
16.1 关系型数据库与非关系型数据库	219	18.6.2 基本操作	281
16.2 Redis 的安装使用	220	18.7 模板	284
16.3 Redis 数据类型	223	18.7.1 模板赋值与变量输出	285
16.3.1 string	223	18.7.2 使用函数和运算符	287
16.3.2 list	227	18.7.3 模板标签	288
16.3.3 hash	233	第 19 章 PHP 设计模式	292
16.3.4 set	236	19.1 什么是设计模式	292
16.3.5 zset	241	19.2 工厂模式	294
16.4 Key 操作命令	246	19.3 单例模式	298
16.5 PHP 操作 Redis	249		
16.5.1 安装 php-redis 扩展	249		
16.5.2 在 PHP 中使用 Redis	251		

19.4 观察者模式	299	22.3 接口开发	355
19.5 策略模式	301	22.3.1 定义路由与封装基类方法	355
第 20 章 基于前端架构打造服务端	303	22.3.2 实现接口功能代码	356
20.1 构建一个 API 的世界	303	第 23 章 编写自己的 PHP 框架	361
20.1.1 简述 API 接口	303	23.1 框架设计	361
20.1.2 API 接口签名验证	304	23.1.1 框架设计思想	361
20.2 传输消息的加解密	305	23.1.2 框架执行流程	361
20.2.1 单向散列加密	305	23.2 框架总体概览	362
20.2.2 对称加密	305	23.3 框架初始化	364
20.2.3 非对称加密	311	23.3.1 入口文件定义	364
20.3 使用 Ajax 进行交互	313	23.3.2 路由解析	365
20.3.1 Ajax 的介绍	313	23.4 应用配置	366
20.3.2 Ajax 的使用	316	23.5 核心基础类	366
20.4 前端模板和框架	318	23.5.1 控制器基类	366
20.4.1 MustacheJS 介绍	318	23.5.2 模型基类	368
20.4.2 AngularJS 介绍	321	23.6 公用函数库	372
第 21 章 O2O 小白财税网站开发	323	23.7 使用框架	372
21.1 需求分析	323	23.7.1 创建表和模型	372
21.2 网站概览	323	23.7.2 创建控制器和视图	373
21.2.1 网站功能	323	第 24 章 大型网站架构设计	376
21.2.2 网站预览	324	24.1 大型网站架构概述	376
21.3 数据库设计	326	24.1.1 大型网站的特点	376
21.3.1 数据库建表	326	24.1.2 大型网站的设计要素	377
21.3.2 连接数据库	329	24.2 负载均衡	378
21.4 使用 ThinkPHP 搭建项目框架	330	24.2.1 负载均衡技术	378
21.4.1 应用目录	330	24.2.2 负载均衡算法	379
21.4.2 引入 PHPMailer 类库	331	24.3 数据库的使用	380
21.4.3 引入 Ping++ 支付模块	333	24.3.1 数据切分	380
21.5 项目代码编写	335	24.3.2 数据库中间件	381
21.5.1 注册登录	335	24.4 消息队列	382
21.5.2 下单购买	342	24.4.1 消息队列的应用场景	383
21.5.3 用户中心	347	24.4.2 消息队列的基本概念	383
第 22 章 卡券管理系统 App 后台开发	353	24.4.3 常用消息队列	384
22.1 PHP 在 App 开发中的应用	353	24.5 网站监控	385
22.2 App 开发中的 JSON 数据	354	24.5.1 数据收集	385
		24.5.2 异常监控与报警	385

第 1 章

走进 PHP 的世界

PHP (Hypertext Preprocessor, 超文本预处理器) 是一种通用开源脚本语言, 其语法吸收了 C 语言、Java 和 Perl 的特点, 利于学习, 使用广泛, 主要适用于 Web 开发领域。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创的语法, 它可以比 CGI 或者 Perl 更快速地执行动态网页。与其他的编程语言相比, PHP 是将程序嵌入到 HTML (标准通用标记语言下的一个应用) 文档中去执行, 执行效率比完全生成 HTML 标记的 CGI 要高许多。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。

2015 年 6 月, 官方发布了 PHP 7 Alpha 1 版本, 同年 12 月 3 日发布 GA 版本, PHP 7 的发布对于 PHP 来说是具有里程碑意义的。在性能上, PHP 7 的执行效率是原来 PHP 5 的两倍左右, 和 HHVM 相当。相对于 PHP 5.6.x, PHP 7 多了以下几个主要的新特性。

- 提升性能: PHP 7 速度是 PHP 5.6 的两倍。
- 支持 64 位。
- 许多重大错误可以进行异常处理。
- 移除了旧的和不支持的 SAPIs 和扩展。
- null 合并操作符 (??)。
- 结合比较运算符 (<=>)。
- 标量类型声明。
- 匿名类。

当然, PHP 7 相对于以前的版本还有很多不同之处, 但是大部分是兼容以前版本的, 所以大多数情况下无须修改代码就可以迁移到 PHP 7。

1.1 快速认识 PHP

本节首先介绍 PHP 语言的优势、运行机制和原理，以及 PHP 7 的新特性，帮助读者对 PHP 有一个初步的认识。

1.1.1 PHP 语言的优势

PHP 语言主要有以下几点优势：

(1) PHP 学习入门快、开发成本低，语法相对简单，并且提供了丰富的类库，如用于图像处理的 GD 库、各种加密扩展（如 OpenSSL 和 Mcrypt 等），可以很方便地直接使用。很多库默认在安装 PHP 环境的时候都是自带的。

(2) PHP 结合 Linux、Nginx 或 Apache、MySQL 可以方便快捷地搭建一套系统，PHP 还支持直接调用系统命令，这样便可以用代码完成许多操作 Linux 的工作，如打包压缩、复制粘贴、重命名、执行 Linux 中 grep 查询筛选等。Nginx 是一个非常优秀的 Web 服务器软件，Nginx 可接收客户端请求，将 PHP 文件发送给 PHP 程序执行，Nginx 中的 PHP 采用 fastCGI 的形式运行脚本。

(3) PHP 支持使用 MySQL、MSSQL、SQLite 等多种数据库，其中与 MySQL 的结合使用最为流行。PHP 提供了 3 种连接 MySQL 的扩展，包括 MySQL 扩展、MySQLi 扩展和 PDO 扩展，MySQL 扩展在 PHP 5.5 及以后的版本中不再支持，MySQLi 是 PHP 推出的专门用于链接 MySQL 的更加安全高效的扩展，并且提供了更高级的一些操作，完全支持面向对象。PDO 扩展是 PHP 推出的链接 MySQL 和其他类型的数据库的一种统一解决方案，可移植性很高，使用它可以灵活方便地切换不同类型的数据库，而不需变动更多的代码。

(4) PHP 是解释执行的脚本语言，写完程序以后可以立即执行，不像 C、Java、C++ 等其他语言需要编译再执行，这使得 PHP 的开发效率更高。

(5) PHP 中使用到的配置文件相对简单，与 PHP 运行有关的配置文件常用的有 php-fpm.conf 和 php.ini 两个，并且配置参数也简单易懂。更改了 PHP 的配置文件不需要重新启动即可继续运行，因为 PHP 每次运行程序前都会主动加在配置文件中，这比 Java 等其他语言方便多了。

(6) PHP 作为最流行、使用最为广泛的 Web 开发语言，有着丰富的生态圈，有许多著名的开源框架可供使用，如官方的 Zend Framework、CakePHP、Yaf、Symfony 等，开源论坛有 Discuz!、PHPwind 等，开源博客 WordPress，开源网店系统如 Ecshop、ShopEx 等，开源的 SNS 系统如 UCHome、ThinkSNS 等。基于这些优秀的开源系统，你可以方便快速地搭建一套 Web 站点。另外，活跃的社区氛围也能帮助你快速解决开发中遇到的问题。

(7) 结合 LVS 负载均衡、消息队列、数据库主从等技术，PHP 能够支持一般大型网站的应用，满足绝大多数场景下的应用开发。

(8) PHP 本身是由 C 语言开发的，在一些对性能有严苛要求的情况下，还可以使用 C 语言编写 PHP 的扩展来提升程序的执行速度，使用 PHP 完成主要业务的代码编写，使用 C 完成性能提升的需求，这使得可以保证软件开发效率的同时兼顾执行效率。在这种对软件开发速度和程序执行性能有极致追求的情况下，如果是其他语言，可能会让你束手无策，或者推倒重来。

(9) 国内的许多大公司,如百度、淘宝、360 等公司都广泛地使用 PHP 作为开发语言,在具体实践中已经取得了很大成功,有许多成功的经验可供借鉴。

1.1.2 PHP 的运行机制和原理

PHP 由内核 Zend 引擎和扩展层组成,PHP 内核负责处理请求、完成文件流错误处理等操作,Zend 引擎可以将 PHP 程序文件转换成可在虚拟机上运行的机器语言,扩展层提供一些应用层操作需要的函数类库等,比如数组和 MySQL 数据库的操作等。

Zend 引擎是用 C 语言实现的,将 PHP 代码通过词法语法解析成可执行的 opcode 并实现相应的处理方法、基本的数据结构内存分配和管理等,对外提供相应的可供调用的 API 方法。Zend 引擎是 PHP 的核心,所有的外围功能都是围绕它实现的。扩展层通过组件的方式提供各种基础服务、内置函数,标准库都是通过它实现的。用户也可以编写自己的扩展来实现特定的需求。服务端应用编程接口(Server Application Programming Interface, SAPI),通过一系列钩子函数使得 PHP 可以和外围交互数据。我们平时编写的 PHP 程序就是通过不同的 SAPI 方式得到不同的应用模式,如通过 WebServer 实现的 Web 应用和在命令行下运行的脚本等。

一段 PHP 程序被执行的时候会先被解析成 opcode 指令,然后在虚拟机中按顺序执行,由于 PHP 本身是用 C 语言开发的,所以其在执行的时候调用的都是 C 的函数。opcode 是 PHP 程序执行的最基本单位。

HashTable 是 Zend 的核心数据结构,实现了 PHP 里几乎所有的功能,支持 key->value 查询,添加删除的复杂度是 $O(1)$,支持线性遍历和混合类型。在 HashTable 中既有 key->value 形式的散列结构,也有双向链表模式,使得它能够非常方便地支持快速查找和线性遍历。Zend 的散列结构是典型的 hash 表模型,通过链表的方式来解决冲突。Zend 的 HashTable 是一个自增长的数据结构,当 hash 表数目满了之后,其本身会动态地以 2 倍的方式扩容并重新布置元素位置,初始大小均为 8。另外,在进行 key->value 快速查找的时候,Zend 本身还做了一些优化,通过空间换时间的方式加快速度。比如在每个元素中都会用一个变量 nKeyLength 标识 key 的长度以做快速判定。Zend HashTable 通过一个链表结构实现了元素的线性遍历。理论上,做遍历使用单向链表就够了,使用双向链表的主要目的是为了快速删除链表元素,避免遍历。

PHP 是一门弱类型语言,本身不严格区分变量的类型。PHP 在声明变量的时候不需要指定类型。PHP 在程序运行期间可能进行变量类型的隐式转换。和其他强类型语言一样,程序中也可以进行显式的类型转换。Zval 是 Zend 中另一个非常重要的数据结构,用来标识并实现 PHP 变量。

Zval 主要由以下 3 部分组成。

- Type 指定了变量所述的类型(整数、字符串、数组等)。
- refcount&is_ref 用来实现引用计数。
- value 是核心部分,存储了变量的实际数据。

Zval 用来保存一个变量的实际数据。因为要存储多种类型,所以 zval 是一个 union,也由此实现了弱类型。

引用计数在内存回收、字符串操作等地方使用得非常广泛。PHP 中的变量就是引用计数的典型应用。Zval 的引用计数通过成员变量 is_ref 和 ref_count 实现。通过引用计数,多个变量可以共享同一份数据,避免频繁复制带来的大量消耗。

在进行赋值操作时,Zend 将变量指向相同的 Zval,同时 ref_count++,在 unset 操作时,对应

的 `ref_count-1`。只有 `ref_count` 为 0 时才会真正执行销毁操作。如果是引用赋值，Zend 就会修改 `is_ref` 为 1。

PHP 变量通过引用计数实现变量共享数据，当试图写入一个变量时，Zend 若发现该变量指向的 `Zval` 被多个变量共享，则为其复制一份 `ref_count` 为 1 的 `Zval`，并递减原 `Zval` 的 `refcount`，这个过程称为“`Zval` 分离”。可见，只有在有写操作发生时，Zend 才进行复制操作，因此也叫 `copy-on-write`（写时复制）。

对于引用型变量，其要求和非引用型相反，引用赋值的变量间必须是捆绑的，修改一个变量就修改了所有捆绑变量。

1.1.3 PHP 7 的新特性

相较于以前的版本，PHP 7 在语言语法层面和底层架构层面都有一些改进。在语法层面的改进主要是增加了一些新特性、移除了一些扩展、改变了错误异常处理等。在底层结构方面，改变了存储各种变量的 `Zval` 和 `Zend_String` 结构体、优化了 `Zend Array` 的 `HashTable`、改进了函数的调用机制等。这些底层结构的改进大幅提升了 PHP 的执行效率，使得其执行速度比 PHP 5 高出一倍左右。

PHP 是一个弱类型的语言，不过在 PHP 7 中支持变量类型的定义，引入了一个开关指令 `declare(strict_type=1);`。这个指令一旦开启，就会强制当前文件下的程序遵循严格的函数传参类型和返回类型。不开启 `strict_type`，PHP 将会尝试转换成要求的类型；开启之后，PHP 不再做类型转换，类型不匹配就会抛出错误。要使用严格模式，一个 `declare` 声明指令必须放在文件的顶部。这意味着严格声明标量是基于文件可配的。这个指令不仅影响参数的类型声明，还影响函数的返回值声明。

PHP 7 中的新特性主要有以下几点：

- (1) 标量类型声明。
- (2) 函数返回值类型声明。
- (3) 新增 `null` 合并运算符。
- (4) 新增组合比较符。
- (5) 支持通过 `define()` 定义常量数组。
- (6) 新增支持匿名类。
- (7) 支持 `Unicode codepoint` 转译语法。
- (8) 更好的闭包支持。
- (9) 为 `unserialize()` 提供过滤。
- (10) 新增加 `IntlChar` 类。
- (11) 支持 `use` 语句从同一 `namespace` 导入类、函数和常量。
- (12) 新增整除函数 `intdiv()`。
- (13) `session_start()` 支持接收数组参数。

除了以上列举的 13 点新特性之外，还有其他一些变更，读者可到 <http://php.net/manual/zh/migration70.new-features.php> 查看有关 PHP 7 新特性的详细变更和示例。

另外，在 PHP 7 中，很多致命错误以及可恢复的致命错误都被转换为异常来处理了。这些异常继承自 `Error` 类，此类实现了 `Throwable` 接口（所有异常都实现了这个基础接口）。

这也意味着，当发生错误的时候，以前代码中的一些错误处理的代码将无法被触发。因为在

PHP 7 版本中，已经使用抛出异常的错误处理机制了。（如果代码中没有捕获 Error 异常，就会引发致命错误）。

在 2013 年的时候，惠新宸和 Dmitry（PHP 语言内核开发者之一）就曾经在 PHP 5.5 的版本上做过一个 JIT（Just In Time，即时编译，一种软件优化技术）的尝试。PHP 5.5 原来的执行流程是将 PHP 代码通过词法和语法分析编译成 opcode 字节码，然后 Zend 引擎读取这些 opcode 指令，逐条解析执行。他们在 opcode 环节后又引入了类型推断（TypeInf），然后通过 JIT 生成 ByteCodes 再执行。采用这种技术优化，PHP 的效率在实际项目中并没有取得明显的提升，于是他们重新设计了 PHP 的底层语言结构。Zval 是存储 PHP 中变量的载体，是一个 C 语言实现的结构体（struct），PHP 5 的 Zval 在内存中占据 24 个字节，而在 PHP 7 中优化后的 Zval 只占 16 个字节，这样变量的存储变得非常简单和高效。PHP 7 优化了数组的 HashTable 实现，PHP 5 的数组存储形式是一个支持双向链表的 HashTable，不仅支持通过数组的 key 来做 hash 映射访问元素，也能通过 foreach 以访问双向链表的方式遍历数组元素。当我们通过 key 值访问一个元素内容的时候，有时需要 3 次的指针跳跃才能找到需要的内容。最重要的一点是，这些数组元素的存储是分散在各个不同的内存区域的，在 CPU 读取的时候，因为它们很可能不在同一级缓存中，导致 CPU 不得不到下级缓存甚至内存区域查找，从而引起 CPU 缓存命中下降，进而增加更多的耗时。优化后的 Zend Array 最大的特点是整块的数组元素和 hash 映射表全部连接在一起，被分配在同一块内存中。如果是遍历一个整型的简单类型数组，效率会非常快，因为数组元素（Bucket）本身是连续分配在同一块内存里的，并且数组元素的 Zval 会把整型元素存储在内部，也不再有点针外链，全部数据都存储在当前内存区域内。当然，最重要的是它能够避免 CPU 缓存命中率下降。另外，PHP 7 还改进了函数的调用机制，通过优化参数传递的环节减少了一些指令，提高执行效率。

1.2 搭建 PHP 开发环境

对于初学者，推荐在 Windows 操作系统下使用 XAMPP 一键安装 PHP 集成开发环境（Apache、PHP、MySQL），XAMPP 提供 PHP 7 的安装版本，读者只需要到官方网站 <https://www.apachefriends.org/download.html> 下载即可。下载界面如图 1-1 所示。

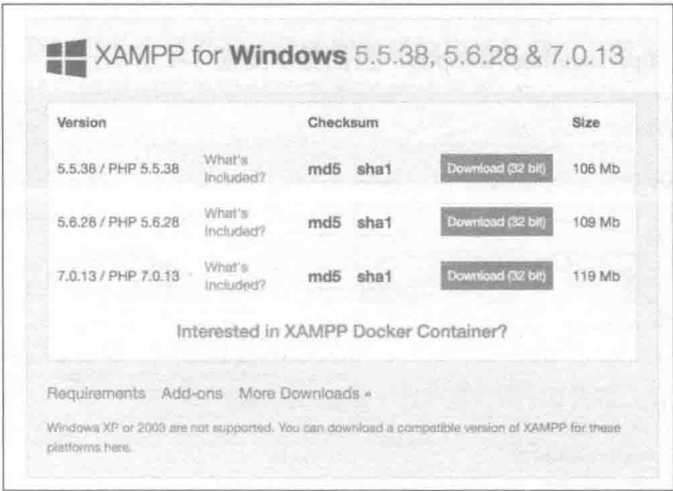


图 1-1 XAMPP 下载页面

下载后得到一个 EXE 文件，双击该文件安装。安装完成后，查看该集成环境安装目录，如图 1-2 所示。

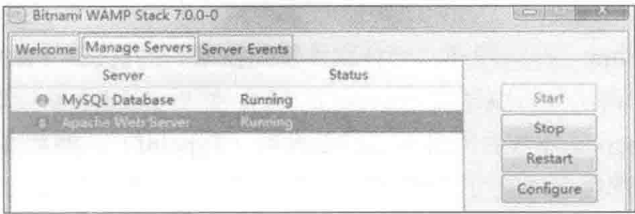


图 1-2 XAMPP 安装目录

双击 manager-windows.exe 即可打开管理窗口，在 Manage Servers 选项卡查看 MySQL 及 Apache 运行状态，如图 1-3 所示。



图 1-3 查看运行状态

应用目录默认位于安装目录的 apache2/htdocs 目录下，在该目录下新建 test.php 并编辑其内容：

```
<?php
echo phpinfo();
```

在浏览器中访问 http://localhost/test.php，页面显示如图 1-4 所示。

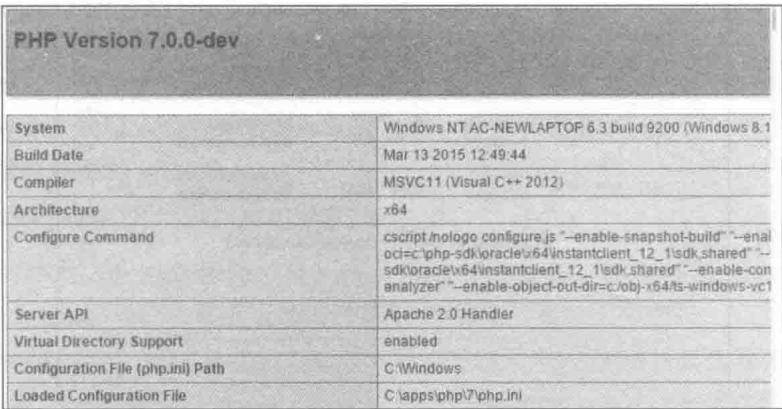


图 1-4 安装成功

1.3 常用 PHP 代码编辑器

下面介绍几个常用的 PHP 代码编辑器。

1. Sublime Text

Sublime 是一个超漂亮的跨平台编辑器，速度快并且功能丰富，有很多可供选择的插件，几乎支持所有的编程语言，支持多行选择、高亮显示、代码缩放、键盘绑定、宏、拆分视图等，同时拥有全屏和免打扰模式。它同时支持 Linux、Windows 和 OSX。这个编辑器可以无限期试用，也可以付费购买。

2. Notepad++

Notepad++是一款免费又优秀的文本编辑器，支持在 Windows 环境下运行的多种编程语言。Notepad++支持超过 50 种编程、脚本和标记语言的语法高亮显示和代码折叠，能让用户迅速减小或扩大代码段，以便查阅整个文档。用户也可以手动设置当前语言，覆盖默认语言。

3. Vim

Vim 编辑器和其他的代码编辑器不同的是命令行的工作方式。和简单的输入代码不同，你选择输入、选择文字，运行正则表达式的搜索，并且使用更多其他的命令。Vim 使用脚本和插件可以变得非常适合扩展，可以支持 GUI 或者命令行，同时可以支持所有的操作系统。在大多数的 Linux 系统都预先安装。

4. PHPStorm

PHPStorm 是一个轻量级且便捷的 PHP IDE，旨在提高用户效率，可深刻理解用户的编码，提供智能代码补全，快速导航以及即时错误检查。在 PHPStorm 里可配置智能的开发环境，VCS 支持 SVN、Git、Mercurial 等，可以连接到数据库，是 PHP 开发者最常用的编辑软件之一。

5. Dreamweaver

Dreamweaver 是第一套针对专业网页设计师特别发展的视觉化网页开发工具，利用它可以轻而易举地制作出跨越平台限制和跨越浏览器限制的充满动感的网页。Dreamweaver 使用所见即所得的接口，亦有 HTML（标准通用标记语言下的一个应用）编辑功能。它有 Mac 和 Windows 系统的版本，随 Macromedia 被 Adobe 收购后，Adobe 也开始计划开发 Linux 版本的 Dreamweaver 了。

1.4 编写一个“Hello World” PHP 程序

上两节学习了安装 PHP 的开发环境，也介绍了几个常用的代码编辑器，这一节我们就来编写第一个 PHP 程序。从最经典的“hello world”开始吧，PHP 代码如下：