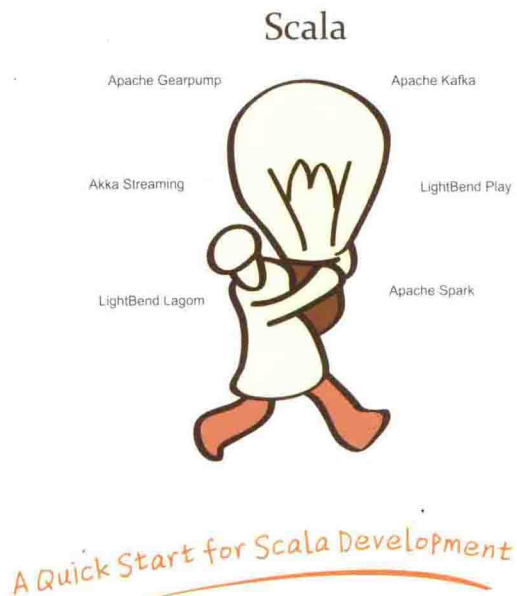


资深大数据工程师开发经验分享，循序渐进地介绍Scala语言的核心语法、深层原理及实际应用
内容丰富，概念阐述清晰，代码注释详尽，可操作性强



Scala

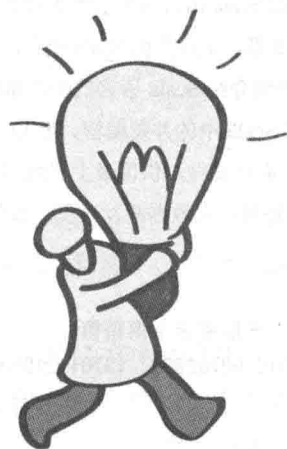
开发快速入门

周志湖 牛亚真 编著

清华大学出版社



本书源代码下载



Scala

开发快速入门

周志湖 牛亚真 编著

清华大学出版社
北京

内 容 简 介

本书以 Scala 语言的核心语法为主线,通过大量实例代码循序渐进地介绍了 Scala 语言的语法特性。第 1 章介绍 Scala 语言及开发环境的安装、IntelliJ IDEA 的使用、Scala 语言的交互式命令行。第 2~3 章重点介绍 Scala 语言的基础语法结构。第 4 章,重点介绍 Scala 语言重要的数据结构 collection (集合)。第 5 章,重点介绍 Scala 语言中面向函数编程的语法特性。第 6~8 章,重点介绍 Scala 语言面向对象编程的语法特性。第 9 章,介绍 Scala 语言中的模式匹配及原理。第 10 章,详细介绍 Scala 语言的类型系统。第 11 章,对 Scala 语言中的隐式转换及原理进行详细介绍。第 12 章,重点介绍 Scala 中的并发编程。第 13 章,介绍 Java 语言与 Scala 语言间的互操作。

本书还提供了所有实例的源代码与素材文件,供读者上机练习使用,读者可从网上下载本书资源文件。

本书适用于 Scala 语言初学者、爱好者,大数据开发人员,软件培训机构以及计算机专业的学生使用。

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

Scala 开发快速入门 / 周志湖, 牛亚真编著. —北京: 清华大学出版社, 2016
ISBN 978-7-302-44413-8

I. ①S… II. ①周… ②牛… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 168661 号

责任编辑: 王金柱

封面设计: 王 翔

责任校对: 闫秀华

责任印制: 王静怡

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: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 190mm×260mm 印 张: 19.75 字 数: 506 千字

版 次: 2016 年 9 月第 1 版 印 次: 2016 年 9 月第 1 次印刷

印 数: 1~3000

定 价: 59.00 元

产品编号: 069160-01

推荐序

蒲晓蓉 博士

经过近两三年的媒体炒作、技术发展及部分行业应用落地，大数据不再悬在空中，而是真正地在影响着你的日常生活，“数据就是资源”、“数据就是石油”、“数据就是金矿”等数据价值理念已深入人心。

IDC 出版的数字世界研究报告显示，2013 年人类产生、复制和消费的数据量达到 4.4ZB，而到 2020 年，数据量将增长 10 倍，达到 44ZB，中国众多的人口数量及广泛应用的硬件基础设施将使我国 2020 年的数据量增长 24 倍，为我国大数据产业的发展提供良好的基础。党和政府为适应经济发展的新常态，稳步推进经济结构调整，助力经济转型升级，2015 年 10 月，党的十八届五中全会公报提出要实施“国家大数据战略”，大数据第一次写入党的全会决议，从此开启了我国大数据建设的新篇章。

回归现实，大数据行业面临着诸如人才稀缺、技术和创业门槛高、行业可借鉴经验少等众多问题，其中人才稀缺是当前最迫切需要解决的问题。虽然基于开源 Hadoop 生态体系的大数据解决方案一定程度上降低了技术门槛和创业门槛，但面临具体应用场景时，开源体系中可选择工具众多，如 SQL-On-Hadoop 工具就有 Hive、Impala、Presto、Stinger 等，这对技术人员的要求非常高，因为不同的工具开发语言、兼容性等可能都不同。Apache Spark 的出现极大地降低了大数据从业人员的门槛，它“一个软件栈内完成所有大数据分析任务（one stack to rule them all）”的思想及基于内存的计算模式一定程度上代表着未来大数据技术的发展方向。

Scala 语言具有简洁、优雅、与 Java 语言的良好互操作性及融合函数式与面向对象编程思想等诸多优点，值得每个优秀的程序员花时间去学习。在大数据领域，除 Apache Spark 底层采用 Scala 语言实现外，还有 Apache Kafka 及 Apache Gearpump 等优秀的大数据处理框架；在其他应用开发领域，还有 Akka 并发编程框架、Lagom 微服务框架、Play Web 应用开发框架等优秀的产品，这已足以证明该语言的强大。

本书作者具备深厚的计算机专业背景、科研能力和写作水平，对新事物充满好奇心，善于学习新知识。通过自学 Scala 语言，积累了丰富的第一手资料和经验，并将个人学习经验通过本书分享出来，有助于 Scala 语言初学者的学习，一定程度上推动国内 Scala 语言的普及。

蒲晓蓉

2016 年 6 月

蒲晓蓉，博士，电子科技大学计算机科学与工程学院教授，英才实验学院副院长。全国计算机学会高级会员；第二届机械工业教育协会会员；电子工业出版社特聘专家；教育部高校教师培训中心特聘专家。出版国家级“十五”、“十一五”规划教材等8部，《计算机操作系统》国家级资源共享课负责人，获四川省教学成果二等奖、四川省科技进步二等奖。主要研究方向为机器学习、计算机视觉和操作系统。

前言

为什么要写这本书

Apache Spark、Apache Kafka 等基于 Scala 语言实现的框架在大数据生态圈内占有举足轻重的地位，它们一定程度上引领着大数据最前沿的技术方向，另外 Akka、Apache Gearpump 等基于 Scala 语言实现的框架在大数据生态圈内也备受关注，工业实践已经证明了 Scala 语言的强大，这使 Scala 语言近一两年名声大噪，进而引起众多编程语言爱好者及各大公司的亲睐，国内外也因此掀起了一场学习 Scala 语言的热潮。

Scala 语言基于 JVM 平台，能够与 Java 语言进行良好地互操作，它最大的特点是可以将面向对象编程语言与函数式编程语言的特点结合起来。Scala 语言的简洁性和表达能力，使得 Scala 语言与 Java 语言相比，同样的功能用 Scala 实现代码量要少很多。Scala 语言的众多优点，让许多业内名家对其推崇备至，这其中便有 Java 之父 James Gosling 和 Groovy 语言创始人 James Strachan。

“If I were to pick a language to use today other than Java, it would be Scala.”^[1]（如果现在让我选择使用除 Java 之外的一门语言的话，那一定是 Scala）——Java 之父 James Gosling

“Though my tip though for the long term replacement of javac is Scala. I'm very impressed with it! I can honestly say if someone had shown me the Programming in Scala book by Martin Odersky, Lex Spoon & Bill Venners back in 2003 I'd probably have never created Groovy.”（我认为将来可能替代 javac 的就是 Scala，它极大地震撼了我，老实说如果有人在 2003 年把 Martin Odersky、Lex Spoon 和 Bill Venners 写的那本《Programming in Scala》书拿给我看了的话，那我可能就不会再去发明 Groovy 语言了）^[2]——Groovy（另一种基于 JVM 平台的语言）创始人 James Strachan。

目前国内 Scala 方面教材的短缺一定程度上限制了 Scala 语言在国内的普及，本书试图弥补这一空白。

本书内容

本书是笔者在 Scala 学习、工作实践及培训过程中的心得体会和系统总结。内容涵盖 Scala 语言基础知识，包括变量的定义、程序控制结构、Scala 集合操作；Scala 语言中级知识，包括 Scala 函数式编程、Scala 面向对象编程、Scala 模式匹配；Scala 语言高级编程知识，包括 Scala 类型参数、隐式转换、Scala 并发编程、Scala 与 Java 互操作。书中利用大量的具体示

¹ <http://www.edureka.co/blog/why-scala-is-getting-popular/>

² <http://macstrac.blogspot.com/2009/04/scala-as-long-term-replacement-for.html>

例和实际案例来说明 Scala 语言的应用，既能够掌握核心语法，又能够理解其背后的深层次原理。

读者对象

(1) Spark、Kafka 等框架二次开发人员

Spark、Kafka 等大数据处理框架目前在稳定性、扩展性方面虽然已经取得了长足的进步，但在实际使用时难免遇到问题，此时需要深入到内核源代码中分析问题，在理解其架构原理与实现细节的基础上通过修改内核源代码来解决问题，这需要开发人员有很强的 Scala 语言功底。

(2) Spark、Kafka 等框架应用开发人员

在学习 Spark、Kafka 等 Scala 语言实现的大数据技术框架时，这些框架大多都提供了 Java、Scala、Python 等上层应用 API 编程接口，但仅仅学会使用其上层 API 是不够的，因为上层 API 抽象程度较高，各框架的底层实现细节、设计原理等对开发人员来说是一个黑匣子，在遇到性能等问题时，开发人员如果对底层实现细节不熟悉的话可能很难进行程序的性能优化。

(3) Spark、Kafka 等框架运维工程师

目前 Spark、Kafka 等大数据处理框架基本上已经成为各大公司大数据解决方案的标配，但基于开源解决方案常常会面临一些技术风险，框架本身的问题及实际应用场景的不同可能会遇到很多故障，因此对于这些使用 Scala 语言实现的框架而言，运维工程师也需要知晓各框架的底层细节，这样才能够迅速定位问题并找到合适的解决办法。

(4) 从事大数据技术的 Java 开发人员

Scala 语言完全兼容 Java 语言，Scala 语言中大量使用了 Java 语言现有的库，因此如果你是一个 Java 大数据开发人员，在此建议学习 Scala 语言，相信它会给你带来全新的感受，甚至有抛弃 Java 的念头。

(5) 大数据技术开源爱好者

Spark、Kafka 等是大数据技术框架中的明星，Scala 语言已经通过了工业界的检验，学习 Scala 语言为学习这些框架的设计原理打下坚实的基础，为以后参与开源、学习工业界最先进的大数据技术架构的优秀思想打下了坚实的基础。

如何阅读本书

本书涵盖三大主要部分：

(1) Scala 语言基础篇，主要介绍 Scala 语言入门基础知识、变量的定义、程序控制结构、Scala 集合。

(2) Scala 语言中级篇，主要介绍 Scala 函数式编程、Scala 面向对象编程、Scala 模式匹配。

(3) Scala 语言高级篇，主要介绍 Scala 类型参数、隐式转换、Scala 并发编程、Scala

与 Java 互操作。

如果你是一名初学者，请按照书的顺序从第一章基础理论部分开始学习，学习时先认真看书中的代码示例和说明，然后照着代码亲自动手实践，这样可以达到事半功倍的学习效果；如果你是一名 Scala 资深用户，你可以自由阅读各章内容，相信书中部分内容肯定能够给你带来一些新的启发。

资源下载

本书代码可以从 <http://pan.baidu.com/s/1nvYDC1r>（注意数字和字母大小写）下载。如果下载有问题，请电子邮件联系 booksaga@163.com，邮件标题为“Scala 开发快速入门-代码”。

勘误和支持

由于笔者水平有限，加之编写时间仓促，书中难免出现一些错误或不准确的地方，恳请读者批评指正。有任何问题，可以关注微信公众号：**ScalaLearning** 或者通过邮箱 403408607@qq.com 联系，编者将尽最大努力及时为读者提供相应的解答。

编 者

2016 年 6 月于杭州

致 谢

感谢我的导师蒲晓蓉教授，在我攻读硕士研究生时，是她给予我项目、学术及为人处事方面的诸多指导，是蒲老师带我进入机器学习、计算机视觉的世界，从而有幸能够接触到 Scala 语言。

感谢清华大学出版社的王金柱老师及他的团队，王老师在近八个月的时间内始终支持我的写作，是他的鼓励和帮助让我能够在繁忙的工作之余孜孜不倦地完成本书的编写。

感谢对本书部分章节提出建议的师兄杨智杰，感谢部门总经理陈霄一直以来对我的信任和工作上的指导，还要感谢赵永标、秦海龙在工作上对我的支持与帮助。

最后，感谢父母及亲朋好友的鼓励和支持，感谢他们在我困顿中给予无私帮助；感谢老婆对我生活的悉心照顾和宽容；感谢我可爱的女儿逸霖，是她的笑容让我忘却烦恼，坚持每天写作。

谨以此书献给我最亲爱的家人和朋友，以及所有热爱 Scala 语言及开源技术的朋友们。

周志湖

2016 年 6 月于杭州

目 录

第 1 章	Scala 入门	1
1.1	Scala 简介	2
1.2	Scala 开发环境搭建	2
1.2.1	软件准备	2
1.2.2	JDK 的安装与配置	3
1.2.3	Scala SDK 的安装与配置	5
1.2.4	IntelliJ IDEA 的安装与配置	6
1.3	Scala Hello World	8
1.3.1	创建 Scala Project	8
1.3.2	配置项目代码目录结构	9
1.3.3	创建应用程序对象	11
1.3.4	运行代码	12
1.4	IntelliJ IDEA 常用快捷键	13
1.4.1	代码编辑类常用快捷键	13
1.4.2	导航快捷键	14
1.4.3	编译、运行及调试	15
1.4.4	代码格式化	15
1.5	交互式命令行使用	15
1.5.1	Scala 内置交互式命令行	15
1.5.2	Scala Console 交互式命令行	17
1.5.3	Scala Worksheet 的使用	19
小 结		20
第 2 章	变量及基本数据类型	21
2.1	变 量	22

2.1.1	变量定义	22
2.1.2	lazy 变量	23
2.2	基本数据类型	24
2.2.1	Int 类型	24
2.2.2	Float 类型	25
2.2.3	Double 类型	25
2.2.4	Char 类型	25
2.2.5	String 类型	26
2.2.6	Boolean 类型	27
2.3	基本类型操作	27
2.3.1	算术运算操作	27
2.3.2	关系运算操作	28
2.3.3	逻辑运算操作	28
2.3.4	位运算操作	29
2.3.5	对象比较运算操作	30
2.3.6	字符串运算操作	31
2.4	运算符的优先级	32
2.5	元组类型	33
2.6	符号类型	34
	小 结	35
第 3 章	程序控制结构	36
3.1	if 语句	37
3.1.1	if 语句	37
3.1.2	if... else... 语句	37
3.1.3	if...else if...else... 语句	37
3.1.4	if 的嵌套使用	38
3.1.5	if 表达式	39
3.2	while 循环语句	39
3.2.1	while 语句的使用	39
3.2.2	do while 语句的使用	40
3.3	for 循环语句	41
3.3.1	基础 for 循环	41
3.3.2	有过滤条件的 for 循环	44

3.3.3 多重 for 循环	44
3.3.4 作为表达式的 for 循环	45
小 结	45
第4章 集 合	46
4.1 集合简介	47
4.2 数 组	49
4.2.1 定长数组	49
4.2.2 变长数组 ArrayBuffer	50
4.2.3 数组的遍历	52
4.2.4 遍历生成数组	53
4.2.5 常用函数	53
4.3.6 多维数组	54
4.3 列 表 (List)	55
4.3.1 列表的创建	55
4.3.2 List 常用函数	56
4.3.3 List 伴生对象方法	58
4.4 集 (Set)	59
4.5 映 射 (Map)	61
4.6 队 列 (Queue)	63
4.7 栈 (Stack)	64
小 结	66
第5章 函 数	67
5.1 函 数	68
5.2 值 函 数	69
5.2.1 值函数的定义	69
5.2.2 值函数的简化	71
5.3 高阶函数	73
5.3.1 高阶函数的定义	73
5.3.2 高阶函数的使用	74
5.4 闭 包	78
5.5 函数柯里化	80
5.6 部分应用函数	81

5.7 偏函数	83
小 结	85
第 6 章 Scala 面向对象编程 (上)	86
6.1 类与对象	87
6.1.1 类的定义	87
6.1.2 创建对象	88
6.1.3 类成员的访问	88
6.1.4 单例对象	90
6.1.5 应用程序对象	91
6.1.6 伴生对象与伴生类	93
6.2 主构造函数	96
6.2.1 主构造函数的定义	96
6.2.2 默认参数的主构造函数	98
6.2.3 私有主构造函数	99
6.3 辅助构造函数	100
6.3.1 辅助构造函数的定义	100
6.3.2 辅助构造函数中的默认参数	102
6.4 继承与多态	104
6.4.1 类的继承	104
6.4.2 构造函数执行顺序	107
6.4.3 方法重写	108
6.4.4 多态	109
6.5 成员访问控制	111
6.5.1 默认访问控制	112
6.5.2 protected 访问控制	113
6.5.3 private 访问控制	114
6.5.4 private[this] 访问控制	116
6.5.5 主构造函数中的成员访问控制	121
6.6 抽象类	125
6.6.1 抽象类的定义	125
6.6.2 抽象类的使用	126
6.7 内部类与内部对象	128
6.8 匿名类	130

小 结	131
第 7 章 Scala 面向对象编程 (下)	132
7.1 trait 简介	133
7.2 trait 的使用	135
7.2.1 trait 的几种不同用法	135
7.2.2 混入 trait 的类对象构造	138
7.2.3 提前定义与懒加载	140
7.3 trait 与类	144
7.3.1 trait 与类的相似点	144
7.3.1 trait 与类的不同点	146
7.4 多重继承问题	147
7.5 自身类型	148
小 结	151
第 8 章 包 (package)	152
8.1 包的定义	153
8.2 包的使用和作用域	155
8.2.1 包的使用	155
8.2.2 包作用域	156
8.3 包 对 象	159
8.4 import 高级特性	160
8.4.1 隐式引入	160
8.4.2 引入重命名	160
8.4.3 类隐藏	161
小 结	162
第 9 章 模式匹配	163
9.1 模式匹配简介	164
9.2 模式匹配的 7 大类型	167
9.2.1 常量模式	167
9.2.2 变量模式	167
9.2.3 构造函数模式	169
9.2.4 序列模式	170

9.2.5 元组模式	171
9.2.6 类型模式	172
9.2.7 变量绑定模式	173
9.3 模式匹配原理	174
9.3.1 构造函数模式匹配原理	174
9.3.2 序列模式匹配原理	176
9.4 正则表达式与模式匹配	177
9.4.1 Scala 正则表达式	177
9.4.2 正则表达式在模式匹配中的应用	180
9.5 for 循环中的模式匹配	183
9.6 模式匹配与样例类、样例对象	185
9.6.1 模式匹配与样例类	185
9.6.2 模式匹配与样例对象	187
小 结	190
第 10 章 隐式转换	191
10.1 隐式转换简介	192
10.2 隐式转换函数	193
10.2.1 隐式转换函数的定义	193
10.2.2 隐式转换函数名称	194
10.3 隐式类与隐式对象	195
10.3.1 隐式类	195
10.3.2 隐式对象	196
10.4 隐式参数与隐式值	197
10.4.1 隐式参数	197
10.4.2 隐式值	199
10.4.3 隐式参数使用常见问题	201
10.5 隐式转换规则与问题	204
10.5.1 隐式转换的若干规则	204
10.5.2 隐式转换需注意的问题	208
小 结	210
第 11 章 类型参数	211
11.1 类与类型	212

11.2 泛 型	214
11.2.1 泛型类	215
11.2.2 泛型接口与泛型方法	216
11.2.3 类型通配符	217
11.3 类型变量界定	220
11.4 视图界定	222
11.5 上下文界定	223
11.6 多重界定	226
11.7 协变与逆变	227
11.8 高级类型	230
11.8.1 单例类型	231
11.8.2 类型投影	235
11.8.3 类型别名	237
11.8.4 抽象类型	238
11.8.5 复合类型	240
11.8.6 函数类型	241
小 结	243
第 12 章 Scala 并发编程基础	244
12.1 Scala 并发编程简介	245
12.1.1 重要概念	245
12.1.2 Actor 模型	247
12.1.3 Akka 并发编程框架	247
12.2 Actor	249
12.2.1 定义 Actor	249
12.2.2 创建 Actor	250
12.2.3 消息处理	255
12.2.4 Actor 的其他常用方法	260
12.2.5 停止 Actor	261
12.3 Typed Actor	265
12.3.1 Typed Actor 定义	265
12.3.2 创建 Typed Actor	266
12.3.3 消息发送	267
12.3.4 停止运行 Typed Actor	269

12.4 Dispatcher	271
12.4.1 常用 Dispatcher	271
12.4.2 ExecutionService	274
12.5 Router	277
12.6 容 错	279
12.6.1 Actor 的 4 种容错机制	279
12.6.2 Supervison	279
小 结	285
第 13 章 Scala 与 Java 的互操作	286
13.1 Java 与 Scala 集合互操作	287
13.1.1 Java 调用 Scala 集合	287
13.1.2 Scala 调用 Java 集合	288
13.1.3 Scala 与 Java 集合间相互转换分析	289
13.2 Scala 与 Java 泛型互操作	290
13.2.1 Scala 中使用 Java 泛型	291
13.2.2 Java 中使用 Scala 泛型	291
13.3 Scala trait 在 Java 中的使用	293
13.4 Scala 与 Java 异常处理互操作	298
小 结	299
参考文献	300