

Broadview
www.broadview.com.cn

PEARSON

快学 Scala

[美] Cay S. Horstmann 著
高宇翔 译

Martin Odersky作序力荐



Scala
for the Impatient



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

快学 Scala

Scala for the Impatient

[美] Cay S. Horstmann 著
高宇翔 译



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Scala是一门以Java虚拟机(JVM)为目标运行环境并将面向对象和函数式编程语言的最佳特性结合在一起的编程语言。你可以使用Scala编写出更加精简的程序,同时充分利用并发的威力。由于Scala运行于JVM之上,因此它可以访问任何Java类库并且与Java框架进行互操作。本书从实用角度出发,给出了一份快速的、基于代码的入门指南。Horstmann以“博客文章大小”的篇幅介绍了Scala的概念,让你可以快速地掌握和应用。实际上手的操作,清晰定义的能力层次,从初级到专家级,全程指导。

本书适合有一定的Java编程经验、对Scala感兴趣,并希望尽快掌握Scala核心概念和用法的开发者阅读。

Authorized translation from the English language edition, entitled Scala for the Impatient, 1E, 9780321774095 by Cay S. Horstmann, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright©2012 Pearson Education Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright ©2012

本书简体中文版专有出版权由Pearson Education培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有Pearson Education培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2012-6207

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

快学Scala / (美)霍斯曼(Horstmann, C.S.)著;高宇翔译. —北京:电子工业出版社,2012.10

书名原文:Scala for the Impatient

ISBN 978-7-121-18567-0

I. ①快… II. ①霍… ②高… III. ①JAVA语言-程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第222314号

策划编辑:张春雨

责任编辑:李云静

印刷:中国电影出版社印刷厂

装订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开本:787×980 1/16 印张:25.5 字数:453千字

印次:2012年10月第1次印刷

定 价:79.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至zllts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010)88258888。

译者序

Scala是一门十分有趣又非常实用的语言，它以JVM为目标环境，将面向对象和函数式编程有机地结合在一起，带来独特的编程体验。

它既有动态语言那样的灵活简洁，同时又保留了静态类型检查带来的安全保障和执行效率，加上其强大的抽象能力，既能处理脚本化的临时任务，又能处理高并发场景下的分布式互联网大数据应用，可谓能缩能伸。

我大约是从2009年开始接触Scala的。在此之前曾做过多年的Java开发，其间也陆续接触过JRuby、Groovy和Python，但没有一门语言能像Scala这样，让我产生持续的兴趣和热情，让我重新感受到学习、思考和解决问题的乐趣。Scala为我开了一扇窗，将我带进了函数式编程的世界，在打破旧有思维模式的同时，让我的整个计算机编程的知识体系重组，看待很多技术问题的角度都不一样了。这种感觉，不亚于早几年前接触Linux。

Scala不光是一门值得用心学习的语言，同时也是一门可以直接上手拿来解决实际问题的语言。它跟Java的集成度很高，可以直接使用Java社区大量成熟的技术框架和方案。由于它直接编译成Java字节码，因此我们可以充分利用JVM这个高性能的运行平台为我们提供的便利和保障。

目前国内外已经有很多公司和个人采用Scala来构建他们的平台和应用。作为JVM上第一个获得广泛成功的非Java语言，Scala正以它独特的魅力吸引着越来越多人的热情投入。

你手里的这本书，出自《Java核心技术》（*Core Java*）的作者，Cay S. Horstmann。每一章的篇幅都不长，娓娓道来，沁人心脾，适合有一定经验的Java程序员阅读。几乎所有Scala相关的核心内容都有涉及，由浅入深，深入浅出，非常适合快速上手。

当然了，如果你想要用好Scala，想把它发挥到更高的层次，基本功必须扎实。这本书讲的都是基本招式，看似平实无华，实则招招受用，对于一线开发人员，实在是

值得放在案头反复揣摩练习。

广大Scala爱好者们，这是为你们准备的书，希望你们也和我一样，在Scala中找到乐趣，找到归属，你们是我完成本书翻译的动力。

感谢Martin Odersky和他的团队，为我们带来如此美妙的编程语言；感谢电子工业出版社、张春雨编辑第一时间从国外引进这本书；感谢老赵、林晴等友人为审稿付出的宝贵时间和精力；感谢编辑团队和其他幕后工作者的辛勤劳动；最后还要感谢家人，感谢你们的理解和支持。

在本书的翻译过程中，译者虽已尽力将原著的真实意思以符合中文习惯的方式呈现给大家，但毕竟能力有限，问题和疏漏在所难免，恳请各位读者批评指正，联系邮箱：gaoyuxiang.scala@gmail.com。

高宇翔

2012年8月于上海

序

几年前我和Cay Horstmann见面，他告诉我Scala需要一本更好的入门书。当时我自己的书才刚出来，因此我当然要问他觉得我那本书哪里不好。他回答说，书很不错，但就是太长了，他的学生们是不会有耐心读完800页的《Scala编程》（*Programming in Scala*）的。我认为他说得有一定道理。然后他就开始着力改变这个状况，于是就有了这本《快学Scala》。

看到这本书终于完成我非常高兴，因为它真切地印证了书名所表达的意思。这是一部快速、实用的Scala入门指引，详细解释了Scala到底有什么特别，与Java的区别在哪里，如何克服学习中常见的困难，以及如何编写优质的Scala代码。

Scala是一门具备高度表达能力且十分灵活的语言。它让类库编写者们可以使用非常精巧的抽象，以便类库的使用者们可以简单、直观地表达自己。因此，根据代码种类的不同，它可以很简单，也可以很复杂。

一年前，我曾试着通过一组用于Scala及其标准类库的层级定义来对这个问题做一些澄清。首先按应用程序开发人员和类库设计者分开，然后各自又分为三个层级。初级的内容可以很快被掌握并且足够用于产出实际代码。中级的内容可以使程序变得更加精简、更加函数式，并且可以让类库使用起来更加灵活。而最高级的内容是为那些解决特定问题、处理特定任务的专家准备的。当时我这样写道：

我希望这个层级划分有助于让Scala的初学者决定以怎样的顺序来学习，并且能够给教师和书籍作者一些建议，以怎样的顺序来呈现相关内容。

Cay的书是第一本系统化地采纳这个想法的Scala入门书。每一章都相应地打上了层级标签，让你一目了然地知道该章的难易程度，以及它是面向类库编写者，还是面向应用程序开发人员。

如你所预期的那样，开始的章节是对基本的Scala功能的快速介绍。不过本书并不

就此收手，接下来还涵盖了许多“高级”概念，直到最后非常高端的内容，这些内容通常并不会出现在编程语言的入门指引当中，比如如何编写解析器组合子，如何使用定界延续，等等。层级标签是一个指引，告诉大家什么时候该看什么内容。Cay令人钦佩地做到了让哪怕是最高级的概念理解起来也那么简单明了。

我非常喜欢本书的构思，于是向Cay和他的编辑Greg Doench询问能否将本书基础章节部分作为免费资料放在Typesafe网站上供大家下载。他们大方地答应了我的请求，对此我深表感谢。这样一来，每个人都可以很快地获取到这份在我看来是目前市面上最紧凑的Scala入门指南。

Martin Odersky

2012年1月

前言

Java和C++的进化速度已经大不如前，那些乐于使用更现代的语言特性的程序员们正在将眼光移向他处。Scala是个很有吸引力的选择；事实上，在我看来，对于想要突破和超越Java或C++的程序员而言，Scala是最具吸引力的一个。Scala的语法十分简洁，相比Java的样板代码，Scala让人耳目一新。Scala运行于Java虚拟机之上，让我们可以使用现成的海量类库和工具。它在拥抱函数式编程的同时，并没有废弃面向对象，使你得以逐步了解和学习一种全新的编程范式。Scala解释器可让你快速运行实验代码，这使得学习Scala的过程颇为轻松惬意。最后，同时也是很重要的一点是，Scala是静态类型的，编译器能够帮我们找出大部分错误，这样一来，这些错误就不至于要等到程序运行起来以后才被发现（或未被发现），造成时间上的浪费。

本书是写给那些对于立即开始Scala编程有急切渴望的读者的。我假定你懂Java、C#或C++，并且我也不会去解释变量、循环或类这些基本概念。我不去穷举Scala的所有特性，不会宣传某一种范式比另一种更优越，也不会用冗长的、过于机巧的示例来折磨你。与此相反，你将会以紧凑的篇幅得到你想要的信息，可以根据需要选择阅读和复习。

Scala是一门内容很丰富的语言，不过你并不需要知道它的所有细节，就已经可以有效地使用了。Scala的创始人Martin Odersky对应用程序开发工程师和类库设计人员所需的专业知识技能的层级进行了定义，如下所示。

应用程序开发工程师	类库设计人员	总体Scala技能层级
初级 A1		初级
中级 A2	初级 L1	中级
专家 A3	高级 L2	高级
	专家 L3	专家

对每一章（偶尔也针对特定的小节），我都标出了所需的经验层级，大致的递进

顺序是：**A1**、**L1**、**A2**、**L2**、**A3**、**L3**。就算你不打算设计你自己的类库，知道Scala为类库设计人员提供了哪些工具，对于更有效地使用类库也会大有裨益。

我希望你通过本书享受到学习Scala的乐趣。如果你发现了错误或者有任何改进建议，请访问 <http://horstmann.com/scala> 并留言。在那里，你也能找到指向包含本书全部代码示例的打包文件的链接。

在此特别感谢Dmitry Kirsanov和Alina Kirsanova将我的手稿从XHTML转换成如此漂亮的排版形式，让我可以将注意力集中在内容而不是在格式调整上。每个作者都应享受这种待遇！

参与本书审稿的人员有：Adrian Cumiskey、Mike Davis、Rob Dickens、Daniel Sobral、Craig Tatoryn、David Walend、William Wheeler。非常感谢你们的评价和建议！

最后，一如既往，感谢我的编辑Greg Doench，感谢他对我撰写本书的鼓励和贯穿于整个过程当中的洞察力。

Cay Horstmann
2012年于旧金山

作者简介

Cay S. Horstmann是《Java核心技术》卷1和卷2第8版（Sun Microsystems出版社2008年出版）的主要作者，除此之外，他还著有其他十多本面向专业程序员和计算机科学专业学生的书籍。他是San Jose州立大学计算机科学专业的教授，同时还是一位Java Champion。

目 录

译者序	III
序	XV
前言	XVII
作者简介	XIX
第1章 基础 A1	1
1.1 Scala解释器	1
1.2 声明值和变量	3
1.3 常用类型	4
1.4 算术和操作符重载	5
1.5 调用函数和方法	7
1.6 apply方法	8
1.7 Scaladoc	9
练习	11
第2章 控制结构和函数 A1	13
2.1 条件表达式	14
2.2 语句终止	16
2.3 块表达式和赋值	16
2.4 输入和输出	17
2.5 循环	18
2.6 高级for循环和for推导式	19
2.7 函数	21

5.3	只带getter的属性	55
5.4	对象私有字段	56
5.5	Bean属性 L1	57
5.6	辅助构造器	59
5.7	主构造器	60
5.8	嵌套类 L1	63
	练习	65
第6章	对象 A1	69
6.1	单例对象	69
6.2	伴生对象	70
6.3	扩展类或特质的对象	71
6.4	apply方法	72
6.5	应用程序对象	73
6.6	枚举	74
	练习	75
第7章	包和引入 A1	79
7.1	包	80
7.2	作用域规则	81
7.3	串联式包语句	83
7.4	文件顶部标记法	83
7.5	包对象	84
7.6	包可见性	85
7.7	引入	85
7.8	任何地方都可以声明引入	86
7.9	重命名和隐藏方法	87
7.10	隐式引入	87
	练习	88
第8章	继承 A1	91
8.1	扩展类	91

8.2	重写方法	92
8.3	类型检查和转换	93
8.4	受保护字段和方法	94
8.5	超类的构造	94
8.6	重写字段	95
8.7	匿名子类	96
8.8	抽象类	97
8.9	抽象字段	97
8.10	构造顺序和提前定义 L3	98
8.11	Scala继承层级	100
8.12	对象相等性 L1	101
	练习	102
第9章	文件和正则表达式 A1	105
9.1	读取行	106
9.2	读取字符	106
9.3	读取词法单元和数字	107
9.4	从URL或其他源读取	108
9.5	读取二进制文件	108
9.6	写入文本文件	108
9.7	访问目录	109
9.8	序列化	110
9.9	进程控制 A2	111
9.10	正则表达式	113
9.11	正则表达式组	114
	练习	114
第10章	特质 L1	117
10.1	为什么没有多重继承	117
10.2	当做接口使用的特质	119
10.3	带有具体实现的特质	120
10.4	带有特质的对象	121
10.5	叠加在一起的特质	122
10.6	在特质中重写抽象方法	124

10.7 当做富接口使用的特质	124
10.8 特质中的具体字段	125
10.9 特质中的抽象字段	126
10.10 特质构造顺序	127
10.11 初始化特质中的字段	129
10.12 扩展类的特质	131
10.13 自身类型 L2	132
10.14 背后发生了什么	133
练习	135
第11章 操作符 L1	139
11.1 标识符	139
11.2 中置操作符	140
11.3 一元操作符	141
11.4 赋值操作符	142
11.5 优先级	142
11.6 结合性	143
11.7 apply和update方法	144
11.8 提取器 L2	145
11.9 带单个参数或无参数的提取器 L2	146
11.10 unapplySeq方法 L2	147
练习	148
第12章 高阶函数 L1	151
12.1 作为值的函数	151
12.2 匿名函数	152
12.3 带函数参数的函数	153
12.4 参数（类型）推断	154
12.5 一些有用的高阶函数	155
12.6 闭包	156
12.7 SAM转换	157
12.8 柯里化	158
12.9 控制抽象	159
12.10 return表达式	161

练习	162
第13章 集合 A2	165
13.1 主要的集合特质	166
13.2 可变和不可变集合	167
13.3 序列	168
13.4 列表	169
13.5 可变列表	170
13.6 集	171
13.7 用于添加或去除元素的操作符	173
13.8 常用方法	175
13.9 将函数映射到集合	177
13.10 化简、折叠和扫描 A3	178
13.11 拉链操作	181
13.12 迭代器	183
13.13 流 A3	184
13.14 懒视图	185
13.15 与Java集合的互操作	186
13.16 线程安全的集合	188
13.17 并行集合	188
练习	190
第14章 模式匹配和样例类 A2	193
14.1 更好的switch	194
14.2 守卫	195
14.3 模式中的变量	195
14.4 类型模式	196
14.5 匹配数组、列表和元组	197
14.6 提取器	198
14.7 变量声明中的模式	199
14.8 for表达式中的模式	199
14.9 样例类	200
14.10 copy方法和带名参数	201
14.11 case语句中的中置表示法	201

14.12 匹配嵌套结构	202
14.13 样例类是邪恶的吗	203
14.14 密封类	204
14.15 模拟枚举	205
14.16 Option类型	205
14.17 偏函数 L2	207
练习	207
第15章 注解 A2	211
15.1 什么是注解	212
15.2 什么可以被注解	212
15.3 注解参数	213
15.4 注解实现	214
15.5 针对Java特性的注解	216
15.5.1 Java修饰符	216
15.5.2 标记接口	216
15.5.3 受检异常	217
15.5.4 变长参数	217
15.5.5 JavaBeans	218
15.6 用于优化的注解	219
15.6.1 尾递归	219
15.6.2 跳转表生成与内联	220
15.6.3 可省略方法	221
15.6.4 基本类型的特殊化	222
15.7 用于错误和警告的注解	223
练习	224
第16章 XML处理 A2	227
16.1 XML字面量	228
16.2 XML节点	228
16.3 元素属性	230
16.4 内嵌表达式	231
16.5 在属性中使用表达式	232
16.6 特殊节点类型	233

16.7 类XPath表达式	234
16.8 模式匹配	235
16.9 修改元素和属性	236
16.10 XML变换	237
16.11 加载和保存	238
16.12 命名空间	241
练习	242
第17章 类型参数 L2	245
17.1 泛型类	246
17.2 泛型函数	246
17.3 类型变量界定	246
17.4 视图界定	248
17.5 上下文界定	249
17.6 Manifest上下文界定	249
17.7 多重界定	250
17.8 类型约束 L3	250
17.9 型变	252
17.10 协变和逆变点	253
17.11 对象不能泛型	255
17.12 类型通配符	256
练习	257
第18章 高级类型 L2	259
18.1 单例类型	259
18.2 类型投影	261
18.3 路径	262
18.4 类型别名	263
18.5 结构类型	264
18.6 复合类型	265
18.7 中置类型	266
18.8 存在类型	267
18.9 Scala类型系统	268
18.10 自身类型	269