HZ BOOKS
华章教育

计 算 机 科 学 丛 书

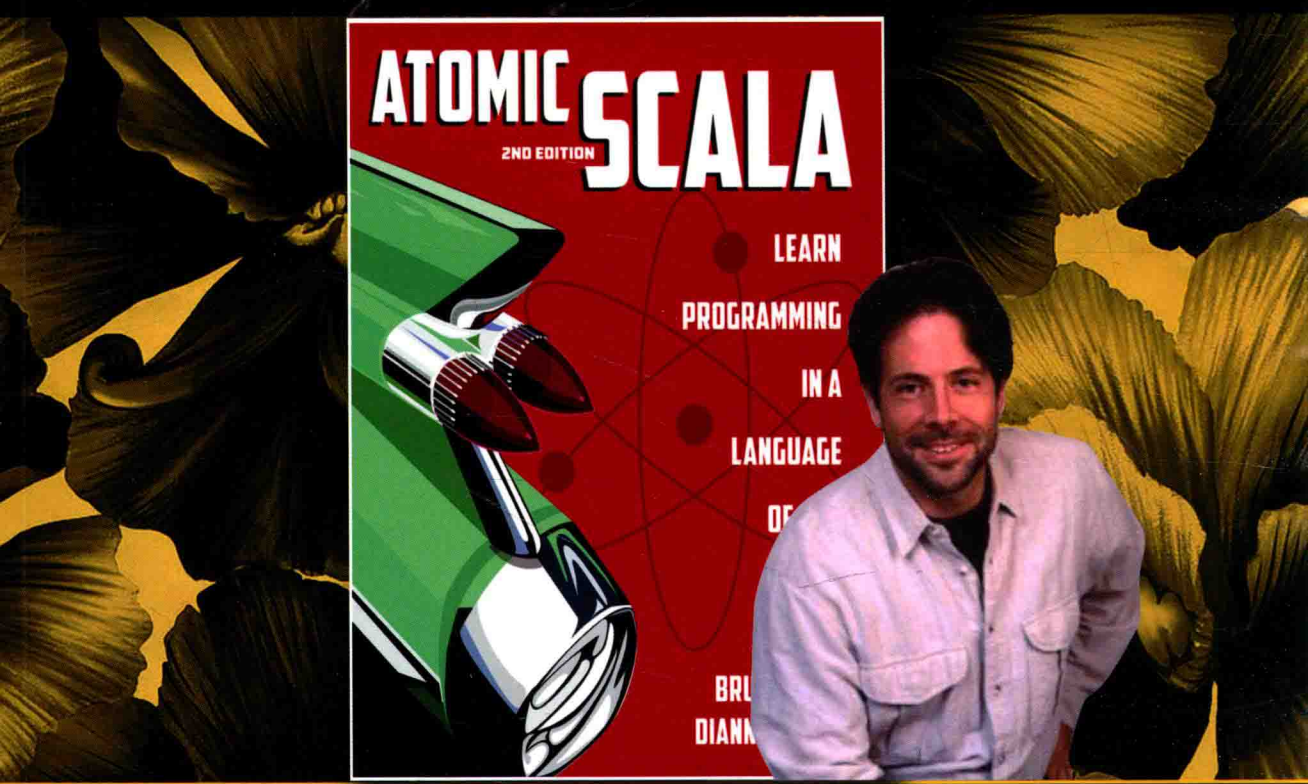
原书第2版

大数据时代掌握Scala编程的基础知识和核心技术的最佳入门指南
程序设计大师Bruce Eckel继《Java编程思想》之后最新力作

Scala编程思想

[美] 布鲁斯·埃克尔（Bruce Eckel） 戴安娜·马什（Dianne Marsh） 著
陈昊鹏 译

Atomic Scala
Learn Programming in a Language of the Future Second Edition



机械工业出版社
China Machine Press

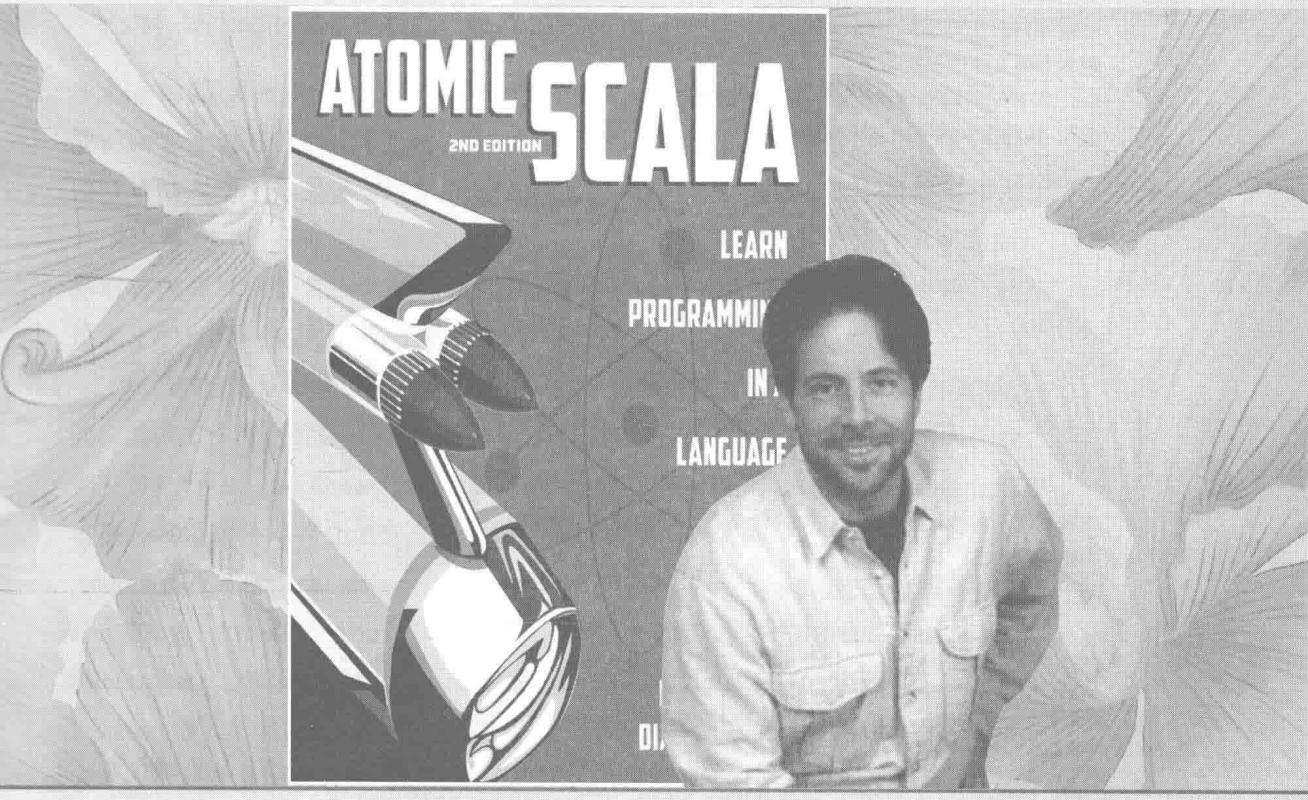
计 算 机 科 学 丛 书

原书第2版

Scala编程思想

[美] 布鲁斯·埃克尔（Bruce Eckel） 戴安娜·马什（Dianne Marsh） 著
陈昊鹏 译

Atomic Scala
Learn Programming in a Language of the Future Second Edition



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Scala 编程思想（原书第 2 版）/（美）埃克尔（Eckel, B.），（美）马什（Marsh, D.）著；陈昊鹏译．—北京：机械工业出版社，2015.9
（计算机科学丛书）
书名原文：Atomic Scala: Learn Programming in a Language of the Future, Second Edition

ISBN 978-7-111-51740-5

I. S… II. ① 埃… ② 马… ③ 陈… III. JAVA 语言－程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2015）第 239713 号

本书版权登记号：图字：01-2015-0846

本书介绍 Scala 的基础特性，采用短小精悍的“原子”解构 Scala 语言的元素和方法。一个“原子”即为一个小型知识点，通过代码示例引导读者逐步领悟 Scala 的要义，结合练习鼓励读者在实践中读懂并写出地道的 Scala 代码。访问 www.AtomicScala.com 可下载练习解答和代码示例，还可了解本书英文版的最新动态。

本书无需编程背景知识，适合 Scala 初学者阅读。同时，本书也为有经验的程序员提供了“快车道”，共同探索编程语言未来的模样。

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：曲　熠　　责任校对：董纪丽
印　　刷：北京诚信伟业印刷有限公司　　版　　次：2016 年 1 月第 1 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：20
书　　号：ISBN 978-7-111-51740-5　　定　　价：69.00 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) 88378991　88361066　　投稿热线：(010) 88379604
购书热线：(010) 68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们 与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家

不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：（010）88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

华章科技图书出版中心

译者序

Java、C#、C++ 等编程语言当今仍然占据着绝对优势的市场份额，但是求知欲和探索欲驱使人们不断思考未来的语言应该是什么样的。人们给出了很多答案，Scala 就是其中之一。很多人都听说过 Scala 代码比 Java 代码更简洁且更灵活，甚至有些 Scala 程序员宣称：“完成相同的功能，我写的 Scala 程序只有你写的 Java 程序的三分之一的代码量。”对此，人们在惊讶之余非常想亲眼看看 Scala 到底多么神奇。

本书是了解 Scala 基础特性的绝佳入门读本，内容结构和文字风格简洁流畅，既适合毫无背景的初学者，又适合经验丰富的程序员，是以 Scala 的特点编写的介绍 Scala 语言的优秀著作。本书配套网站（www.AtomicScala.com）还提供了大量实用材料，包括练习解答和相关活动信息。

Scala 语言本身博大精深，作为初级读本，本书只涵盖了基础特性，并未涉及高级特性（例如函数式编程等内容）。即便如此，读者也能在阅读本书后顺利开启面向对象编程之旅，并且为了解 Scala 的高级特性做好准备。

在翻译本书时，译者尽力做到在确保准确的情况下使译文更加流畅且更符合中文表达习惯，但由于水平有限，离“信达雅”的标准可能还有差距，欢迎读者批评指正。

陈昊鹏

2015 年 8 月

ATOMIC SCALA
Learn Programming in a Language
of the Future, Second Edition

前言

这应该是你的第一本有关 Scala 的书，而不是最后一本。我们呈现的内容将足以使你熟知这门语言并感到得心应手——你将掌握这门语言，但还不足以成为专家。通过阅读本书，你将编写出有用的 Scala 代码，但是不必追求读懂碰到的所有 Scala 代码。

读完本书后，你就可以阅读更加复杂的 Scala 书籍了，在本书的末尾我们推荐了几本。

这是一本为新手准备的专用书籍。之所以称为“新手”，是因为本书并不要求你之前具备编程知识，而“专用”是因为书中包含丰富的内容，足够自学成才。我们给出了有关编程和 Scala 的基础知识，但是并没有用这门语言博大精深的完整知识体系来淹没你。

属于初学者的程序员应该将其看作一个游戏：你可以通关，但是需要一路解决多个难题。有经验的程序员能够快速阅读本书，并且发现需要慢下来留心阅读的地方。

原子概念

所有编程语言都是由各种特性构成的，运用这些特性可以产生运行结果。Scala 非常强大：它不仅有更多特性，而且可以通过大量不同的方式来表示这些特性。如果我们将这些特性和表示方式一股脑地抛给你，你肯定会觉得 Scala“过于复杂”，从而放弃学习。

然而不必如此。

如果了解这些特性，那么你就可以阅读任何 Scala 代码，并且梳理出其中的含义。事实上，对于一整页的 Scala 代码，如果用其他语言编写具有相同效果的代码则需要许多页，因而理解 Scala 代码显得更容易，因为只需“一页”

就可以看到所有代码。

为了避免揠苗助长，我们会遵循下面的原则循循善诱地教授这门语言：

1. 积跬步以至千里。我们抛弃了将每一章都编写成长篇大论的做法，取得代之的是将每一小步都表示成“原子性”概念，或者简称“原子”，它们看起来就像微缩的章。典型的原子包括一个或多个可运行的小型代码段以及它们产生的输出。我们将描述哪些特性是 Scala 的创新和独到之处，并且努力做到每个原子只表示一个新概念。

2. 无任何前向引用。对作者而言，这种描述方式很有用：“这些特性将在后续章节中进行阐述。”这会使读者发懵，所以我们不会这么做。

3. 无任何对其他语言的引用。我们几乎从来不引用其他语言 (只在绝对必要时才引用)。我们不知道你已经掌握了哪些语言 (如果有的话)，如果我们用某种你不理解的语言的某个特性来进行类比，那么肯定会挫伤你的积极性。

4. 事实胜于雄辩。与纯粹用文字来描述特性不同的是，我们更喜欢用示例和输出来说明特性。通过阅读代码来了解特性显然更好。

5. 实践出真知。我们设法首先展示语言的机制，然后再解释为什么会有这些特性。这种做法似乎落后于“传统”教学方式，但往往更有效。

我们努力工作以期创造最好的学习体验，但是仍然要提醒你：为了易于理解，我们偶尔会过度简化或抽象某个概念，而你之后可能会发现这个概念不完全正确。我们并非经常这么做，凡是这么做都是经过深思熟虑的。我们相信这样做有助于使现在的学习更轻松，并且一旦你了解了详情，就会适应这种方式。

交叉引用

当我们引用本书中的另一个原子时，会为该原子加上底纹，例如，欢迎阅读原子类和对象。

如何使用本书

本书的读者对象既包括编程初学者，也包括已经学会使用其他语言编程的程序员。

初学者。从前言开始，像读其他书一样顺序阅读每个原子，包括“总结”原子，总结内容有助于巩固所学知识。

有经验的程序员。因为你已经理解了编程的基础知识，所以我们为你准备了“快车道”：

1. 阅读前言。

2. 按照相应原子中介绍的方式在你的平台上安装必要的软件。我们假设你已经安装过某种程序编辑器，并且会使用 shell，否则，请阅读编辑器和 shell。

3. 阅读运行 Scala 和编写脚本。

4. 跳到总结 1，阅读其内容并解答其中的练习。

5. 跳到总结 2，阅读其内容并解答其中的练习。

6. 至此，从模式匹配开始，继续按照正常方式通读本书。

第 2 版中的修订

第 2 版中的修订大多是源于 bug 报告的小修改和订正，以及针对 Scala 2.11 版本而做的必要更新。另外还对相当数量的拖沓冗长的行文进行了精简。如果你买过第 1 版的电子书，那么将会自动获得第 2 版的更新。如果你买过第 1 版的纸质书，那么可以在 AtomicScala.com 网站上找到第 2 版中的所有修订。

本书样章

为了更好地介绍本书并引领你进入 Scala 的世界，我们发布了免费的电子版样章，你可以在 AtomicScala.com 上找到。我们尽力让样章足够长，使得它自身就非常有用。

无论是纸质版还是电子版，本书完整版都是需要付费的。如果你喜欢免费样章中所呈现的内容，那么请支持我们，通过付费帮助我们继续完成更多工作。我们希望本书对你有所帮助，并且非常感激你的资助。

在互联网时代，控制任何信息看似都是绝无可能的。你也许能够在许多地方找到本书的完整电子版，如果你此刻无力支付，因而从某个网站上下载了它，那么就请你“将知识传播出去”。例如，在你学会 Scala 之后帮助他人学习 Scala，或者只是以急他人所急的方式帮助他们。也许在未来的某天，风光起来的你会乐于慷慨解囊。

示例代码和练习解答

这些都可以在 AtomicScala.com 下载。

咨询

Bruce Eckel 认为咨询要想上境界，其基础是理解团队或组织的特定需求

和能力，并基于这种理解发现能够以最佳方式将你扶上马走一程的工具和技术。这包括在多个领域内的指导和协助：帮助你分析计划，评估能力和风险，辅助设计，工具评估和选择，语言培训，项目引导研讨会，开发过程中的指导性访问，指导性的代码走查，以及特定主题的研究和现场培训。要想了解 Bruce 是否能够为你的需求提供合适的咨询服务，请通过 MindviewInc@gmail.com 联系他。

会议

Bruce 组织了一个空间开放的会议 Java Posse Roundup(现已成为一个冬季技术论坛，www.WinterTechForum.com)，以及另一个针对 Scala 的同样秉承空间开放原则的会议 Scala Summit(www.ScalaSummit.com)。Dianne 组织了 Ann Arbor Scala Enthusiasts group，同时她还是 CodeMash 的组织者之一。加入 AtomicScala.com 邮件列表，就会收到我们的活动和演讲通知。

支持我们

撰写本书及其各类辅助材料可是一个大项目，这花费了我们大量的时间和精力。如果你喜欢本书，并且想看到更多类似的精品，那么就请支持我们吧：

- 写博客或发 tweet 等，并转发给你的好友。这是一种草根式的拓展市场行为，因此你所做的任何事都会有助于本书的推广。
- 在 AtomicScala.com 购买本书的电子版或纸质版。
- 在 AtomicScala.com 浏览其他辅助产品或 App。

关于我们

Bruce Eckel 是获得多项大奖的《 Thinking in Java 》和《 Thinking in C++ 》的作者，他还创作过大量有关计算机编程的其他书籍。他在计算机产业界已经耕耘了 30 余载，不断地经历着这样的循环：感到挫败，尝试退出，然后诸如 Scala 这样的新生事物产生，带来新的希望，又将他拉回老本行。他在世界各地做了成百上千场报告，并且乐于参加像冬季技术论坛和 Scala Summit 之类的各种会议和活动。Bruce 住在科罗拉多州的 Crested Butte，他经常在当地社区剧院中表演。尽管他此生可能最多也就是个中级滑雪健将或山地车手，但是他认为这些活动和画抽象画一样，都是人生中不可或缺的部分。Bruce 拥有应用物理专业的学士学位以及计算机工程专业的硕士学位。他目前正在学习组

织动力学，以期找到组织公司的新方式，使一起工作变成一种乐趣。你可以在 www.reinventing-business.com 上阅读他在组织方面的奋斗事迹，而他在编程方面的工作可以在 www.mindviewinc.com 上找到。

Dianne Marsh 是 Netflix 云工具工程部门 (Engineering for Cloud Tools) 的主管。她是 SRT Solutions 的创始人之一，这是一家客户软件开发公司，在 2013 年被出售之前，公司一直由她负责运营。她的专长是编程和技术，包括制造、基因组学决策支持和实时处理应用系统。Dianne 在职业生涯伊始使用的是 C，后来喜欢的语言包括 C++、Java 和 C#，目前她非常喜欢 Scala。Dianne 协助组织了 CodeMash (www.codemash.org)，这是一个全部由志愿者构成的开发者大会，使用各种语言的开发者齐聚一堂并彼此学习。她还是 Ann Arbor Hands-On Museum 的董事会成员。她积极参加本地用户组，并且主持着其中的好几个。她在密歇根技术大学 (Michigan Technological University) 获得计算机科学硕士学位。Dianne 嫁给了她最好的朋友，养育了两个可爱的孩子。就是她说服了 Bruce 撰写本书。

致谢

我们感谢 Programming Summer Camp 2011 的参与者对本书的早期评论和参与，特别感谢 Steve Harley、Alf Kristian Stoyle、Andrew Harmel-Law、Al Gorup、Joel Neely 和 James Ward，他们都慷慨地奉献了自己的时间和评论。还要感谢许多对本书 Google Docs 格式进行评阅的人。

Bruce 要感谢 Josh Suereth 为本书提供的所有技术帮助。还要感谢 Crested Butte 的 Rumors Coffee and Tea House 和 Townie Books，他在撰写本书时在这两家店里花了不少时间。还有 Bliss Chiropractic 的 Mimi 和 Jay，在写作过程中，他们总是定期帮他把事情都理顺。

Dianne 要感谢她在 SRT 的业务搭档 Bill Wagner，以及 SRT Solutions 的雇员，因为她占用了他们工作之外的时间。她还要感谢 Bruce，因为他同意与她一起撰写本书，并在此过程中一直对她不离不弃，尽管他肯定已经因她所犯的被动语态和标点符号错误而身心俱疲。她还要特别感谢丈夫 Tom Sosnowski，感谢他在写作过程中给予的宽容和鼓励。

最后，感谢 Bill Venners 和 Dick Wall，他们的“通向 Scala 的天梯”（Stairway to Scala）课程帮助我们巩固了对这门语言的理解。

题献

献给 Julianna 和 Benjamin Sosnowski。你们无与伦比。

版权

注：书中页边标注的页码为英文原书页码，与索引中的页码一致。

ATOMIC SCALA
Learn Programming in a Language
of the Future, Second Edition

目录

编辑器

ATOMIC SCALA: Learn Programming in a Language of the Future, Second Edition

要想安装 Scala，你可能需要对系统配置文件做些修改，而这就需要称为编辑器的程序。你还需要用编辑器来创建 Scala 程序文件，即本书中所展示的代码清单。

编程用的编辑器从集成开发环境（IDE，例如 Eclipse 和 IntelliJ IDEA）到单机程序，种类繁多。如果已经装有 IDE，那么就用它来编写 Scala 吧。但是我们对保持简洁情有独钟，因此在研讨会和展示会上使用的都是 Sublime Text 编辑器，你可以在 www.sublimetext.com 找到它。

Sublime Text 在所有平台（Windows、Mac 和 Linux）上都可以运行，并且拥有内置的 Scala 模式，当你打开 Scala 文件时，就会自动调用该模式。它并非重型 IDE，因此不会因功能太多而适得其反，这对我们的目标而言再理想不过了。另一方面，它有一些非常方便的编辑特性，你肯定会喜欢的。可以访问它的网站了解更多细节。

尽管 Sublime Text 是商业软件，但是只要你喜欢，还是可以免费使用（你可能会周期性地碰到请你去注册的弹出窗口，但是这并不妨碍继续使用）。如果你像我们一样，那么你很快就会决定（在经济上）支持他们一下。

还有许多其他的编辑器，它们自身显得小众一些，人们甚至会对它们的优缺点进行激烈争论。如果你发现自己更喜欢某款编辑器，那么改用这款编辑器也不会太困难。重要的是选择一款编辑器，并且用得顺手。

shell

ATOMIC SCALA: Learn Programming in a Language of the Future, Second Edition

如果你之前没有编程经历，那么可能从来没用过操作系统上的 shell（在 Windows 中称为命令行）。shell 回溯到了计算早期的年代，那时做任何事都是通过键入命令实现的，而计算机也是通过打印响应来应答的——所有东西都是基于文本的。

尽管在图形化用户界面的年代它看起来可能显得很原始，但是使用 shell 仍可完成数量惊人的很有价值的工作，我们将经常使用它，既作为安装过程的一部分，又用来运行 Scala 程序。

启动 shell

Mac：点击 Spotlight（在屏幕右上角的放大镜图标），然后键入“terminal”。点击看起来像一个小电视屏幕的应用（或敲击“Return”键），这样就会在你的主目录中启动一个 shell。

Windows：首先启动 Windows 资源管理器，然后导航到你的目录下。在 Windows 7 中，点击屏幕左下角的“开始”按钮，在开始菜单的搜索框区域内键入“explorer”，然后按下“Enter”键。在 Windows 8 中，按下 Windows+Q，键入“explorer”，然后按下“Enter”键。

一旦 Windows 资源管理器开始运行，就可以通过鼠标双击计算机上的文件夹导航到所需的目录。现在，点击资源管理器窗口顶部的地址栏，键入“powershell”，并按下“Enter”键。这将在目标目录中打开一个 shell（如果 Powershell 没有启动，请访问 Microsoft 的网站，从那里下载并安装它）。

为了在 Powershell 中执行脚本（这是测试本书示例时必须做的），你必须首先修改 Powershell 的执行策略。

在 Windows 7 中，转到“控制面板”…“系统与安全”…“管理工具”，右击“Windows Powershell Modules”，并选择“以管理员身份运行”。

在 Windows 8 中，使用 Windows+W 键进入“设置”，选择“应用”，然后在编辑框中键入“power”，点击“Windows Powershell”，然后选择“以管理

员身份运行”。

在 Powershell 提示符处运行下面的命令：

```
Set-ExecutionPolicy RemoteSigned
```

如果出现确认提示，那么通过键入表示“是”的“Y”来确认你想要修改执行策略。

从现在开始，在任何新打开的 Powershell 中，都可以通过在 Powershell 提示符处键入 ./ 以及后面跟着的文件名来运行 Powershell 脚本（就是以 .ps1 结尾的文件）。

Linux：按下 ALT-F2，在弹出的对话框中，键入“gnome-terminal”，然后按“Return”。这将在主目录中打开一个 shell。

目录

目录是 shell 的基本元素之一，用来存放文件以及其他目录。你可以把目录看作带有分支的树。如果 books 是系统中的一个目录，并且有两个作为分支的其他目录，例如 math 和 art，那么我们就说 books 目录是带有 math 和 art 两个子目录的目录。我们将使用 books/math 和 books/art 来引用这两个子目录，因为 books 是它们的父目录。

19

基本的 shell 操作

本节展示的 shell 操作在各个操作系统上都大体一致。下面是在 shell 中可以执行的最精华的操作，也是我们将在本书中用到的操作。

- **变换目录**：使用 `cd`，后面跟着想要跳转到的目录名，或者使用 `cd..` 表示想要跳转到上一层目录。如果想要跳转到一个新目录，同时又想要记住你是从哪里跳出的，那么就使用 `pushd`，后面跟着新目录的名字。然后，仅需使用 `popd`，就可以跳转回前一个目录。
- **目录清单**：`ls` 将显示当前目录下的所有文件和子目录名。还可以使用通配符 *（星号）来缩小搜索范围。例如，如果想要列出所有以 .scala 结尾的文件，那么就可以输入 `ls*.scala`。如果想要列出所有以 F 开头和以 .scala 结尾的文件，那么就可以输入 `ls F*.scala`。
- **创建目录**：使用 `mkdir`（make directory）命令，后面跟着想要创建的目录名。例如，`mkdir books`。