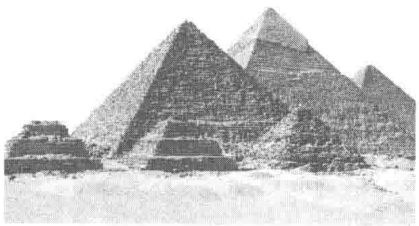
- WebDriver在各种测试场景中的应用
- HTML 5和移动测试，包括原生App和Web App的测试
- 如何进行技术选型并在实际工作中使用BDD
- 使用持续集成工具Jenkins提高团队的测试效率

Selenium
自动化测试之道

一线测试团队经验分享，让初入行的测试人员少走弯路

Ping++测试团队 编著

清華大學出版社



Selenium
自动化测试之道

Ping++测试团队 编著

清華大學出版社
北京

内 容 简 介

本书以 Selenium 的使用为主线，展现了 UI 自动化测试的各种实践过程，引导读者思考如何基于 Selenium 做好 UI 自动化测试。示例代码采用 Python 和 Java，全书共 8 章，第 1 章分析讨论了自动化测试的意义，旨在使读者对自动化测试有一个较明确的认识；第 2、3 章详细介绍了 Selenium IDE 的命令、Selenium WebDriver API、不同 Driver 对象以及工作原理，旨在使读者对 Selenium 有深入的了解；第 4 章重点通过代码演示介绍了不同类型的测试框架；第 5、6 章是拓宽思路，演示了如何使用 Selenium WebDriver 结合 JavaScript 代码来操作 HTML 5 页面的 Web Storage、Canvas 对象，以及如何使用 Appium 处理原生 App 和 Web App 的页面对象；第 7 章着重演示了主流 BDD 框架 Cucumber-JVM、Lettuce、Behave 的应用，偏实战场景，探讨了 BDD 实施过程中需要考虑的种种问题；第 8 章介绍了测试人员在 Jenkins 使用过程中的必备知识。本书还提供了所有示例的源码与素材文件供读者练习使用，读者可从网上下载本书资源文件。

本书适用于具有编程基础，希望系统地了解 UI 自动化测试的开发或测试人员，以及对自动化测试感兴趣的计算机专业学生等。

图书在版编目（CIP）数据

Selenium 自动化测试之道/Ping++测试团队编著. —北京：清华大学出版社，2017
ISBN 978-7-302-48594-0

Ⅰ. ①S… Ⅱ. ①P… Ⅲ. ①软件工具－自动检测 Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字（2017）第 253350 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： 三河市铭诚印务有限公司
经　　销： 全国新华书店
开　　本： 180mm×230mm　　**印　　张：** 13　　**字　　数：** 291 千字
版　　次： 2017 年 11 月第 1 版　　**印　　次：** 2017 年 11 月第 1 次印刷
印　　数： 1~3000
定　　价： 59.00 元

产品编号：071048-01

推荐序一

很开心看到子腾和她的团队的新书即将出版，子腾在学生时期就非常勤奋、务实，非常爱学习、爱钻研问题，给我留下了非常深刻的印象。

参加工作后的子腾在测试行业工作数年，积累了丰富的经验，这本书的主角Selenium就是她非常钟爱的软件测试工具之一。本书的读者覆盖面非常宽泛，你可以没有自动化测试基础，本书第1章就是为这部分读者准备的。此外，这一章还对几个常见且容易混淆的概念进行了解释，例如自动化是否就是白盒测试、自动化和手工测试的比较等，通过一些生动的举例，给刚从事测试工作的读者做了一次概念普及和辨析，非常生动、清楚。

有了开始的概念铺垫以后，第2章开始导入Selenium，这个名字的读音是[sə'li:niəm]，是中文“硒”的意思，它是一套基于 Web 浏览器自动化测试的框架，本章假设读者需要具备一些 Web 的基本概念和基础知识，并至少了解一门编程语言，什么语言并不重要，思想是相通的。本章从Selenium的历史讲起，涵盖安装、使用、实践，并穿插讲述Selenium框架的几个组成部分，建议认真阅读本章内容，采取精读的方式，边读边动手实践，这很重要，给阅读后面的章节打下牢固的基础。通过学习第2章的内容，你可以掌握Selenium的基本使用方法，可以在自己的项目中小试牛刀，当然这并不够，还需要继续阅读。

第3章重点讲述了Selenium WebDriver，它是Selenium框架的核心，也是Selenium适用广泛测试场景的基础。例如，它通过不同的Driver支持主流的浏览器（Firefox、Opera、Safari、IE、Chrome等），也支持没有图形界面的Headless浏览器，掌握了WebDriver，可以让你在各种测试场景中游刃有余，磨刀不误砍柴工，这一章也建议认真阅读。

第5章和第6章对HTML 5和移动测试进行了专题介绍，这也切合了当下的技术发展情况，HTML 5如火如荼，移动化也势不可挡。这两章对HTML 5的基础知识进行了讲解，还需要进一步了解的读者可以自行阅读其他更专业的书籍。移动测试作者讲得更加细致，介绍了Appium以及Appium测试环境从搭建到使用的各个环节，并分别讲述了如何测试iOS和Android移动应用，涵盖原生App和Web App的测试，相信关注移动App测试的读者会收获颇多。

在第 7 章，作者对 BDD（行为驱动测试）进行了专题讲解，BDD 更加注重功能和场景。本章介绍了 BDD 相关的工具，并介绍了如何进行技术选型，找到适合自己的工具。有了合适的工具，就需要学习如何实施了，本章的后半部分重点讲述实际工作中如何使用 BDD 工具，这部分内容读者可以现学现用，直接用到当前的测试工作中。

本书的最后，作者对测试之后的工作进行了延伸，讲述了开源框架 Jenkins，可以提高团队测试效率，建议测试团队的 leader 好好阅读这一部分。

本书的风格一如子腾的性格——严谨、务实，值得想要了解 Selenium 测试框架、想要了解自动化测试的读者认真学习和阅读。

《C＃权威指南》作者姜晓东

2017 年 6 月，南昌

推荐序二

我和子腾最初是在一个测试群里认识的，可谓一见如故，然后我们就经常就测试的各种问题进行讨论，无论我们彼此的观点是否相同，都讨论得很尽兴，有一种“酣畅淋漓”的感觉，不知不觉间，我们成了一对挚友。后来，我知道子腾和她的小伙伴们准备出书，我就自然而然地成了“早鸟”，基本见证了此书从提纲到初稿，再到定稿的整个过程，也见证了子腾和她的小伙伴们为此付出的巨大的心血和努力。

和子腾聊这本书时，才知道原来她在 2014 年，就在网上讲过 Selenium 相关的课程，但这本书绝不是之前网课的文字版，也不是 Selenium 的使用说明书，而是她多年自动化测试沉淀下来的经验，全是那些从网上找不到的内容。也许本书并不能帮读者深入细致地去了解 Selenium 的每个细节，但本书能教会如何才能做好自动化，启发我们在做自动化时，除了考虑框架、技术外，还需要考虑些什么。所以本书和市面上其他讲述 Selenium 的书不同，除了基础内容、WebDriver 外，还有设计模式、BDD、Jenkins 持续集成等能够把自动化在产品中落地，并且有效用起来的内容。

我本人有很多次做自动化的失败经历，就是现在正在做的自动化项目，也是在忐忑中缓慢进行，所以我深知自动化测试要想在实际项目中达到预期效果的不易。读子腾的这本书，从第一章开始，就感到很接地气。关于自动化的老生常谈，虽是“老生”，但谈的都是那些典型、常见的问题，很多问题，我自己也曾经困惑过，如果你是一位自动化测试或者软件测试的新手，这部分内容一定可以帮你解答心中的疑惑。对学习一项新技术来说，最好的方法就是“使用”。读者只需要按照 Selenium 初体验中的描述，Step by Step，就可以快速入门。WebDriver 是掌握 Selenium 必须要理解的内容，本书也花了大量篇幅来描述相关内容，这部分内容也是我最喜欢的内容之一，写得非常翔实，按照本章的指引和演练，读者应该可以写出基本的 Web 自动化脚本，完成部分自动化测试工作了。

很多介绍 Selenium 的书，到这里可能就结束了，但自动化的本质就是用一段代码来测试另一段代码，自动化脚本稳定可靠是自动化测试的基本，另外要想最大程度地发挥自动化的作用，脚本就要尽可能多地被执行，脚本的可移植性从某种程度上来说，甚至超越了产品本身，所以好的自动化测试一定是需要悉心设计的。设计模式这章就是为

了提高自动化脚本的稳定性而编写的。除此之外，书中还介绍了 HTML5 和移动 App 的测试，这些技术在当前都很流行，可以帮助读者丰富自身的自动化测试技术，提升自动化测试实战的应对能力。

事实上，自动化测试要想在项目中发挥好作用，开发模式、流程都是要考虑的因素，特别是对那些使用敏捷方法论的项目来说，自动化变得尤为重要，也往往是测试团队的能力短板。我想为大家推荐本书的一个重要原因就是本书对 BDD、持续集成也进行了系统深入的分析和讨论，这也是本书的一大特色。这样读者就可以把自动化测试做到敏捷项目里，让自动化测试能够发挥更大的作用。

我认识很多优秀的测试工程师，但是能够做到行文流畅简洁，可以说是凤毛麟角，而子腾就是其中之一。阅读本书，你一点也不会感到是种负担，有一种娓娓道来的感觉，犹如一股清泉，但那看似波澜不惊的表面，隐含的却是作者独到的见解和切身体会，这也正是子腾和她的小伙伴们始终在自动化测试领域孜孜不倦研究的结果。我想，对所有热爱测试和渴望技术的人来说，这都是一部可读性很强的作品。阅读它，定会收获满满，不会让你失望。

刘琛梅

2016 年 11 月于蓉

前　言

写一本关于 Selenium 自动化测试的工具书，一开始我是拒绝的。直到现在，我仍然认为工具书不足以道尽测试的奥妙。学习 Selenium 最好的途径是啃官方文档和源码，从最开始的 Selenium RC 到 WebDriver，再到移动测试 Appium，Selenium 一直在快速、持续地发展和变化着。等读者看到这本书的时候，很可能某些问题已经有了更好的解决方案，或者书中的代码已经不能直接运行。

而最终，我还是动笔了。因为我还有另一个观点：“自动化测试”不是某一家公司或者团队组织需要考虑的问题，它应该是测试同行们的必经之路，是日常测试工作的手段之一。而初学者在一开始难免会有畏难情绪，又不知如何构建知识体系。于是，将所思所得分享出来，或许可以帮助初学者尽快地度过那段“破冰期”。

本书的组织方式

市面上 Selenium 的资料很多，谈论测试自动化的也很多。但脱离了工具和技术，去谈方法论，难免让人觉得空洞；而没有方法论的东西，只谈工具和技术，难免是“一叶障目，不见泰山”。本书尝试在梳理技术知识的同时，讨论测试自动化的方法论。

第 1 章主要探讨测试价值观，阐述编者对自动化测试的基本观点和认识。

第 2 章是 Selenium 入门内容，介绍了 Selenium 的发展，涉及 Selenium IDE、Selenium WebDriver 和 Selenium Grid。

第 3 章重点介绍了 Selenium WebDriver 的使用。不是简单罗列 Selenium WebDriver API，还包括不同 WebDriver 对象、不同页面元素的处理思路。

第 4 章介绍了自动化测试框架的设计，包括线性、模块化、数据驱动和关键字框架 4 种类型。

第 5 章介绍了 HTML 5 元素的处理。Selenium 还未对某些 HTML 5 元素的操作进行封装，因而需要利用 JavaScript 来解决问题。读者将在这一章开拓视角，了解更多的 Selenium 应用场景。

第 6 章介绍了移动 App 的测试框架——Appium。基于前面几章对 Selenium 原理与操作的了解，读者会在这一章了解 iOS 与 Android App 自动化测试脚本的写法。

第 7 章介绍了行为驱动开发（BDD）模式。通过这一章，希望读者能体会到做好自动化测试不仅在于工具的掌握和框架的使用，还需要考虑测试用例的管理、手动测试用例如何与自动化脚本关联，甚至与业务部门的沟通等问题，其中几个 BDD 框架的示例为读者提供了解决问题的思路。

第 8 章介绍了持续集成工具 Jenkins 的使用，希望通过这一章能为读者带来测试流程方面的思考。Jenkins 可以让测试脚本的执行、报告的展示变得简单高效。

本书的内容均是由 Ping++的一线测试人员编写的。第 2 章由王红兴、周淼淼编写，第 4 章由徐克亮编写，其余章节由吴子腾编写。

本书的特色

本书的特色主要体现在以下 3 个方面：

第一，在理论观点上，本书在开篇就阐明了编者对于“质量与自动化测试的关系”，“自动化测试与白盒测试的关系”等话题的理解。其实 Selenium 等各种自动化测试工具上手并不难，但相信读者在阅读过程中并不仅仅只是想了解一种工具，而是想获得如何实施自动化测试的思路。正所谓，测试技术或工具只是“指月之手”，我们追求的是“月亮”，是如何放心地迭代，快速地交付高品质的产品。

第二，在学习视角上，本书从 Selenium 工作原理、测试脚本的组织方式——开始讲解，再由 Web 自动化脚本的编写延伸到 HTML 5 元素、App 测试对象的识别等。章节的内容设置与当今企业，尤其是互联网公司所需的 UI 自动化测试技术环环相扣，归纳总结了可能遇到的难点以及解决问题的思路。

第三，在技术实施上，突出了需要向团队传播质量意识与测试自动化实践相结合。本书介绍的行为驱动开发（BDD）与持续集成工具 Jenkins 都是需要团结整个研发团队，甚至是相关的业务部门，才能将这些理念发挥至最佳。当然，即便这些概念在组织推进过程中存在困难，测试人员也可以通过了解这些工具和技术，对研发过程改进这一话题进行更加深入的思考。

考虑到本书的目标和定位，对于没有掌握任何一门编程语言的读者而言，或许会造成阅读门槛。另外，本书涉及多类界面对象的识别和操作、多种测试脚本的写法、多个测试框架的使用。然而在实际工作中，界面操作的自动化仅仅是分层测试策略中的一部分，并不能代表全部的自动化工作。但为了便于从整体上把握和安排内容，编者还是以 Web 测试自动化作为本书的主要架构。这样，相比单一地通过某个系统或产品来整体介

绍自动化测试方面的研究，书中各章节的内容显得在体系性上有所欠缺。

目标读者

本书主要面向的读者是具备编程基础，缺乏自动化测试经验，希望快速、系统地了解 Selenium，从而进一步做好 UI 测试自动化的工程师。本书不仅是为测试人员而写的，它还适用于对软件测试有兴趣的在读大学生以及希望了解测试技术的开发人员。

全书综合了 Selenium 实践过程中的方方面面，涉及脚本编写、框架选型、开发模式等各个领域的讨论。虽然示例代码分为 Java 与 Python 两种语言，但并不会影响阅读，书中对示例代码进行了详尽的文字解读。Python 代码适用于 2.7.10 版本。代码下载链接：https://github.com/applewu/selenium-exercises.git。

如何阅读本书

本书的前 3 章是全书内容的基础，需要首先阅读。在掌握了前 3 章之后，读者可以按照任意顺序阅读后续章节。既可以顺序浏览，概观 Selenium 自动化测试实践，也可以选择性地阅读自己感兴趣的章节。

我们学习任何测试工具的最终目的不在于掌握工具，而在于如何利用工具更好地为自动化测试服务。自动化测试也只是产品质量工作中的一部分。因此，不要沉迷于“术”，而忘却了“道”。在阅读过程中，读者一方面需要积极实践，掌握测试脚本的编写方法，另一方面需要积极思考，如何在自己所在的工作中合理应用起来。练习与反思，才能将本书的效果发挥至极致。

勘误和支持

由于水平有限，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。在阅读过程中遇到任何问题或错误，欢迎发送邮件至邮箱 test4greenbar@163.com ，期待能够得到读者的真挚反馈。

读者还可以直接在 Github 的 selenium-exercises 项目中提交代码有关的问题，也可以通过微博（@籽藤_上海）联系编者。

致谢

首先要感谢清华大学出版社提供了这样一个创作平台。其次，感谢那些提供了宝贵建议的朋友们。虽然最终编写这本书的是 Ping++的测试团队，但还有很多同事和好友为本书提供了宝贵的意见。感谢李雨洪、方雷、孙兵兵、叶波光、翁旭锋、李响、左文娅、赵海林、付敏芝、史子飞提出的问题和反馈，感谢我素未谋面却志同道合的好友刘琛梅以及我的老师姜晓东在百忙之中为本书写了推荐序。

最后，我要感谢我的家人。感谢我的父母，尤其是我的母亲，培养了我的阅读和学习习惯。感谢我的公公婆婆，他们的辛勤付出让我在写书的过程中没有后顾之忧，不用担心儿子的生活起居。我还要感谢我的儿子垲兴，你的笑容是我的能量。感谢你们伴我前行。

Ping++测试团队　吴子腾
上海 张江高科
2017 年 9 月 10 日

目　录

第 1 章

自动化测试的价值观

在行业迅速发展的今天，编写功能测试自动化脚本正逐渐成为测试人员必不可少的技能之一。然而，对于自动化测试在项目或团队中的实施，仍有不少人存在误解，投入了大量时间和精力，结果却事倍功半。

作为全书的开篇，本章先围绕几个常见话题，结合项目实践过程中的所思所得阐明测试价值观。

1.1 自动化测试与产品质量的关系

测试人员思考最多的问题恐怕就是如何才能发现更多更有价值的 Bugs，如何才能更好地避免产品质量上的风险。谈到自动化测试这个话题，出于职业本能，人们往往会第一时间想到：自动化测试如何保证产品质量？

别着急，在提供结论之前，需要有追本溯源的过程。面对这样的问题，先想想看：测试如何保证产品质量？

对于测试与质量问题的关系存在两种极端认识：一种是“测试无用论”，认为开发人员就可以搞定所有测试工作，不需要专业测试人员；另一种认为“待测产品的所有问题都应该被测试人员发现”。

对持有这两种观点的人，笔者都不能苟同。我们可以把测试人员比作医生，拿医生

治病来举例。第一种人否定测试人员的价值，好比此人只是小病小疼，自己吃点药就搞定了，不需要看医生；又或者这人接触过的大多是庸医，根本没有提供过有效的帮助，以至于他对医生失去信心。第二种人把产品质量的全部责任都压在测试人员头上，好比他以为医生是万能的，能发现他身上所有的问题。

正因为没有放之四海而皆准的测试，测试人员本身也面临不少误解，那么对于“自动化测试如何保证产品质量”这种问题就更要留个心眼。笔者非常认同 Cem Kaner 教授的观点，“软件测试是一种技术调查，目的是向相关干系人提供产品相关质量的实验信息”（http://testingeducation.org/wordpress/）。测试人员不是“质量卫士”，测试既不会提高质量，也不会降低质量。尽管有不少公司会把测试人员称为 QA（Quality Assurance），直译就是“质量保证”，但质量是构建出来的，不是测试人员测出来的。因此，测试人员不能保证产品质量，质量保证应当来源于整个产品团队。

既然通过测试不能保证质量，靠自动化测试更无法保证。

综上所述，关于自动化测试保证产品质量的说法，本身就是个伪命题。

那么，应该如何看待自动化测试与质量的关系？

Cem Kaner 的观点给了我们不少启发，笔者认为测试是一项服务性的工作，测试人员在经过一系列信息收集、技术调查后，应当对产品提供质量反馈。而合理有效的自动化测试能够快速获得反馈，从而帮助产品快速迭代。这正是自动化测试的价值所在。

本书在介绍测试技术之余，还在第 7 章介绍了 BDD 方法，第 8 章介绍了 Jenkins 使用的知识，意在强调只有团队协作才能让自动化测试的价值最大化。至于如何保证质量，则属于“质量管理”这一领域的内容，涉及质量目标的制定和指标框架的搭建等方法论，这里就不展开讨论了。

1.2　自动化并不等同于白盒测试

网上有不少介绍白盒测试，分享白盒测试工具的文章。阅读之后会发现，那些并不是白盒测试范畴的内容，只是某种黑盒测试的自动化工具用到了一些编程技术罢了。当大量这类文章充斥着我们的视野，不少测试同行会心生疑惑：自动化就是白盒测试吗？

让我们先通过维基百科上的介绍来梳理一下“黑盒测试”与“白盒测试”的概念。

“黑盒测试，软件测试的主要方法之一，也可以称为功能测试、数据驱动测试或基于规格说明的测试。测试者不了解程序的内部情况，不需具备应用程序的代码、内部结构和编程语言的专门知识，只知道程序的输入、输出和系统的功能，这是从用户的角度针对软件界面、功能及外部结构进行测试，而不考虑程序内部逻辑结构。测试案例是依

应用系统应该实现的功能，照规范、规格或要求等设计的。测试者选择有效输入和无效输入来验证是否正确地输出。此测试方法适用于大部分的软件测试，如集成测试（integration testing）和系统测试（system testing）。”

“白盒测试（white-box testing）又称透明盒测试（glass box testing）、结构测试（structural testing）、逻辑驱动测试或基于程序本身的测试等，软件测试的主要方法之一。测试应用程序的内部结构或运作，而不是测试应用程序的功能（即黑盒测试）。在进行白盒测试时，以编程语言的角度来设计测试案例。测试者输入数据，验证数据流在程序中的流动路径，并确定适当的输出，类似测试电路中的节点。测试者了解待测试程序的内部结构、算法等信息，这是从程序设计者的角度对程序进行的测试。”

通过上述内容可以看出，**自动化测试与白盒测试没有必然联系，它们是不同维度的概念。**是否是白盒测试，要看在设计测试用例、准备测试数据的过程中，是否考虑了待测程序的代码实现逻辑。如果仅凭待测程序的输入输出进行测试，不关心程序的实现细节，那就是黑盒测试，与你选择了哪种自动化测试工具，使用了哪种框架进行测试一点关系也没有。

举个例子，不少购物应用在最后结算时都会根据当前优惠活动对订单自动减价。这是一个很常见的功能，买了 100 元钱，活动优惠了 20 元，最后付款 80 元。对于这种金额计算的测试，你或许不用了解开发程序的实现细节，脑子里就已经浮现出 N 个测试用例了。整数金额、浮点型金额、正常值、等价类、边界值、优惠金额比订单金额大、正交表、并发测试，你的想法越来越多。按照这种思路整理出多个测试场景，把不同的输入值和期望结果整理为测试用例。为了提高下一个版本的测试效率，你写了脚本，不再需要通过手动配置应用的后台金额来进行测试，你的脚本也对期望结果做了充足的验证。于是，测试脚本把你从手动执行的烦琐工作中解脱出来，之后的回归测试还因此发现了几个 Bugs。

当你看着测试脚本，满怀成就感，嘴角微微上扬的时候，你应该意识到，这只是做了黑盒测试自动化。无论你的测试脚本写得多优雅，此时的待测程序对你而言，还是一个黑盒子，测试的出发点并没有考虑“盒子”的内部结构。

你不满足现有的测试用例，开始研究开发代码，试图了解中间数据的存储、计算方式。发现页面上显示的金额是浮点型，单位是“元”，数据库中存储了整型，单位是“分”。Python 代码大概是这样的：

```
# original_price 页面上显示的原订单金额
# discount 数据库中存储可优惠的金额
original_price_db =original_price*100
final_price = original_price_db - discount
print(final_price)
```

是不是很简单？但如果你有浮点型数据的处理经验，你会看出其中的猫腻。

让我们针对上述逻辑整理出两个用例，做一个小实验，如表 1-1 所示。

表 1-1　两个用例说明

用例编号	场景说明
Test-1	original_price = 69.10 discount = 6910
Test-2	original_price = 10.5 discount = 1050

再把上述代码写到 test.py 文件中，观察脚本执行的结果。如图 1-1 所示，当金额为 69.10 元时，若订单金额全免，则最终订单价格不是零。正确做法应该如图 1-2 所示。

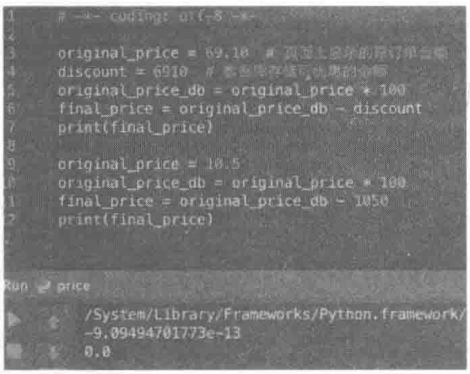


图 1-1　错误的计算语句

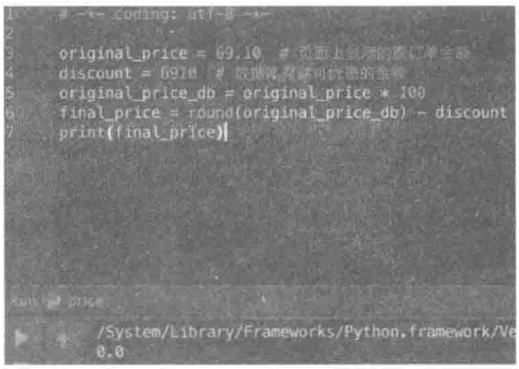


图 1-2　正确的计算语句

这个例子或许不太恰当，合格的开发人员不会犯这种低级错误。但笔者想强调的是，你要意识到测试数据不充分，开始针对代码逻辑设计测试用例。在这一过程中，你的测试方法和策略都是围绕着待测程序的内部逻辑展开的，所以你做的是白盒测试。

请不要把测试自动化与白盒测试等同起来，它们既不是对等也不是对立的关系。

1.3　采用自动化还是手工测试

在求职网站上搜索软件测试的职位，同一个业务领域，同样的从业年限，自动化测试的薪酬普遍会比手工测试高。不少求职者在面试的时候告诉笔者，他的职业规划是做一两年手工测试，之后转做自动化测试；又或者，面试者会告诉笔者，他不想做手工测试，只想做自动化测试。