

O'REILLY®

TURING

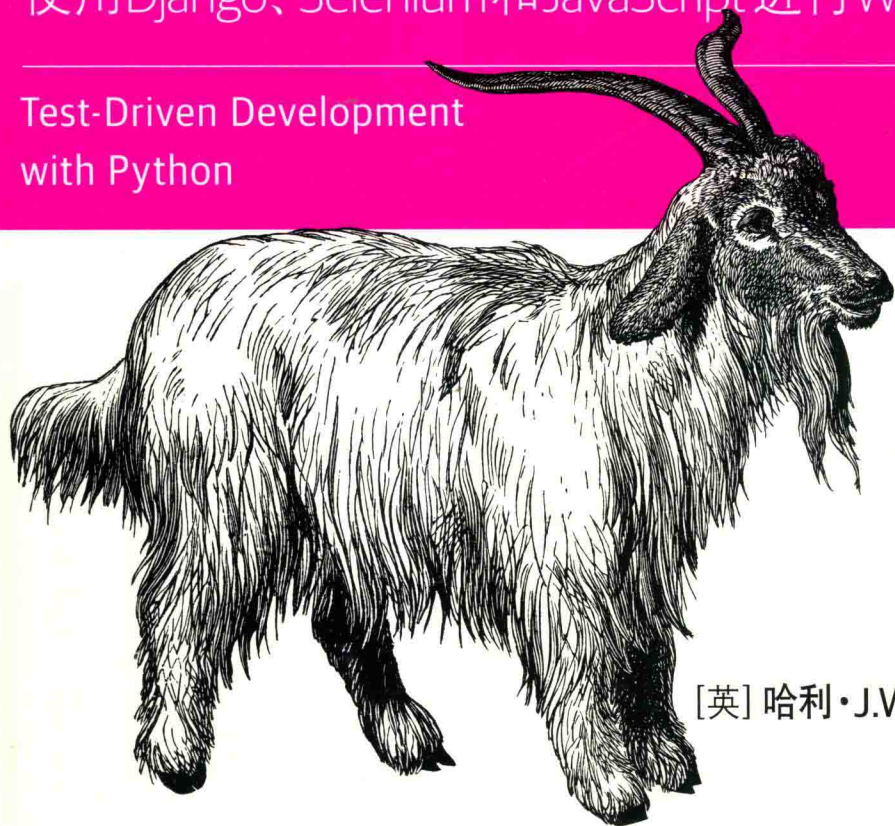
图灵程序设计丛书

第2版

Python 测试驱动开发

使用Django、Selenium和JavaScript进行Web编程

Test-Driven Development
with Python



[英] 哈利·J.W. 帕西瓦尔 著
安道 译



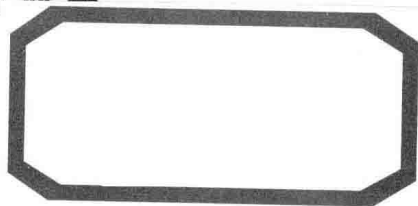
中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计



Python测试驱动开发： 使用Django、Selenium和JavaScript 进行Web编程（第2版）

Test-Driven Development with Python,
Second Edition

[英] 哈利·J.W. 帕西瓦尔 著
安道 译

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北 京

图书在版编目(CIP)数据

Python测试驱动开发：使用Django、Selenium和
JavaScript进行Web编程：第2版 / (英) 哈利·帕西瓦
尔(Harry J. W. Percival) 著；安道译. — 北京：
人民邮电出版社，2018.7

(图灵程序设计丛书)

ISBN 978-7-115-48557-1

I. ①P… II. ①哈… ②安… III. ①软件工具—程序
设计 IV. ①TP311.561

中国版本图书馆CIP数据核字(2018)第111743号

内 容 提 要

本书从最基础的知识开始，讲解 Web 开发的整个流程，展示如何使用 Python 做测试驱动开发。本书由三个部分组成。第一部分介绍了测试驱动开发和 Django 的基础知识，并在每个阶段进行严格的单元测试。第二部分讨论了 Web 开发要素，探讨了 Web 开发过程中不可避免的问题，以及如何通过测试解决这些问题。第三部分探讨了一些高级话题，如模拟技术、集成第三方认证系统、Ajax、测试固件以及持续集成等。

第2版全部使用 Python 3，并针对新版 Django 全面升级，介绍了由外而内的测试驱动开发流程。本书适合 Web 开发人员阅读。

-
- ◆ 著 [英] 哈利·J.W. 帕西瓦尔
 - 译 安 道
 - 责任编辑 朱 巍
 - 执行编辑 夏静文
 - 责任印制 周昇亮
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市祥达印刷包装有限公司印刷
 - ◆ 开本：800×1000 1/16
 - 印张：30.25
 - 字数：721千字 2018年7月第1版
 - 印数：1-3 000册 2018年7月河北第1次印刷
 - 著作权合同登记号 图字：01-2018-3460号
-

定价：119.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广登字 20170147 号

译者介绍

安道

专注于现代计算机技术的自由翻译，译有《流畅的 Python》《Flask Web 开发》《Ruby on Rails 教程》等。



微信连接



回复“Python”查看相关书单



微博连接

关注@图灵教育 每日分享IT好书



QQ连接

图灵读者官方群I：218139230

图灵读者官方群II：164939616

图灵社区
iTuring.cn

在线出版，电子书，《码农》杂志，图灵访谈

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

版权声明

© 2017 by Harry Percival.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2018. Authorized translation of the English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2017。

简体中文版由人民邮电出版社出版，2018。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

我试图通过这本书与世人分享我从黑客变成软件工程师的过程。本书主要介绍测试，但很快你就会发现，除此之外还有很多其他内容。

感谢你阅读本书。

如果你购买了本书，我十分感激。如果你看的是免费在线版，我仍然要感谢你，因为你确定这本书值得花时间来阅读。谁知道呢，说不定等你读完之后，会决定为自己或朋友买一本纸质书。

如果你有任何评论、疑问或建议，希望你能写信告诉我。你可以通过电子邮件直接和我联系，地址是 obeythetestinggoat@gmail.com；或者在 Twitter 上联系我，我的用户名是 @hjwp。你还可以访问本书的网站和博客 (<http://www.obeythetestinggoat.com/>)，以及邮件列表 (<https://groups.google.com/forum/#!forum/obey-the-testing-goat-book>)。

希望阅读本书能让你身心愉悦，就像我在写作本书时感到享受一样。

为什么要写一本关于测试驱动开发的书

我知道你会问：“你是谁，为什么要写这本书，我为什么要读这本书？”

我至今仍然处在编程事业的初期。人们说，不管从事什么工作，都要历经从新手到熟手的过程，最终有可能成为大师。我要说的是，我最多算是个熟练的程序员。但我很幸运，在事业的早期阶段就结识了一群测试驱动开发 (Test-Driven Development, TDD) 的狂热爱好者，这对我的编程事业产生了极大影响，让我迫不及待地想和所有人分享这段经历。可以说，我很积极地做出了最近这次转变，而且这段学习经历现在还历历在目，我希望能让初学者感同身受。

我在开始学习 Python 时（看的是 Mark Pilgrim 写的 *Dive Into Python*），偶然知道了 TDD 的概念。我当时就认为：“是的，我绝对知道这个概念的意义所在。”或许你第一次听说 TDD 时也有类似的反应吧。它听起来像是一个非常合理的方案，一个需要养成的非常好的习惯——就像经常刷牙。

随后我做了第一个大型项目。你可能猜到了，有项目就会有客户，有最后期限，有很多事情要做。于是，所有关于 TDD 的好想法都被抛诸脑后。

的确，这对项目没什么影响，对我也没影响。

但只是在初期如此。

一开始，我知道并不真的需要使用 TDD，因为我做的是个小网站，手动检查就能轻易测试出是否能用。在这儿点击链接，在那儿选中下拉菜单选项，就应该有预期的效果，很简单。编写整套测试程序听起来似乎要花费很长时间，而且经过整整三周成熟的代码编写经历，我自负地认为自己已经成为一名出色的程序员了，我能顺利完成这个项目，这没什么难度。

随后，项目变得复杂得可怕，这很快暴露了我的经验不足。

项目不断变大。系统的不同部分之间要开始相互依赖。我尽量遵守良好的开发原则，例如“不要自我重复”（Don't Repeat Yourself, DRY），却被带进了一片危险地带。我很快就用到了多重继承，类的继承有八个层级深，还用到了 `eval` 语句。

我不敢修改代码，不再像以前一样知道什么依赖什么，也不知道修改某处的代码可能会导致什么后果。噢，天呐，我觉得那部分继承自这里，不，不是继承，是重新定义了，可是却依赖那个类变量。嗯，好吧，如果我再次重定义以前重定义的部分，应该就可以了。我会检查的，可是检查变得更难了。网站中的内容越来越多，手动点击变得不切实际了。最好别动这些能运行的代码，不要重构，就这么凑合吧。

很快，代码就变得像一团麻，丑陋不堪。开发新功能变得很痛苦。

在此之后不久，我幸运地在 Resolver Systems 公司（现在叫 PythonAnywhere）找到了一份工作。这个公司遵循极限编程（Extreme Programming, XP）开发理念。他们向我介绍了严密的 TDD。

虽然之前的经验的确让我认识到自动化测试的好处，但我在每个阶段都心存疑虑。“我的意思是，测试通常来说可能是个不错的主意，但果真如此吗？全部都要测试吗？有些测试看起来完全是在浪费时间……什么？除了单元测试之外还要做功能测试？得了吧，这是多此一举！还要走一遍测试驱动开发中的‘测试/小幅度代码改动/测试’循环？太荒谬了！我们不需要这种婴儿学步般的过程！既然我们知道正确的答案是什么，为什么不直接跳到最后一步呢？”

相信我！我审视过每一条规则，给每一条捷径提出过建议，为 TDD 的每一个看似毫无意义的做法寻找过理由，最终，我发现了采用 TDD 的明智之处。我记不清在心里说过多少次“谢谢你，测试”，因为功能测试能揭示我们可能永远都无法预测的回归，单元测试能让我避免犯很愚蠢的逻辑错误。从心理学上讲，TDD 大大降低了开发过程中的压力，而且写出的代码让人赏心悦目。

那么，让我告诉你关于 TDD 的一切吧！

写作本书的目的

我写这本书的主要目的是要传授一种用于 Web 开发的方法，它可以让 Web 应用变得更好，

也能让开发者更愉快。一本书如果只包含一些上网搜索就能找到的知识，那它就没多大的意思了，所以本书不是 Python 句法指南，也不是 Web 开发教程。我希望教会你的，是如何使用 TDD 理念，更加稳妥地实现我们共同的神圣目标——简洁可用的代码。

即便如此，我仍会从零开始使用 Django、Selenium、jQuery 和 Mock 等工具开发一个 Web 应用，不断提到一个真实可用的示例。阅读本书之前，你无须了解这些工具。读完本书后，你会充分了解这些工具，并掌握 TDD 理念。

在极限编程实践中，我们总是结对编程。写这本书时，我设想自己和以前的自己结成对子，向以前的我解释如何使用这些工具，回答为什么要用这种特别的方式编写代码。所以，如果我表现得有点儿屈尊俯就，那是因为我不是那么聪明，我要对自己很有耐心。如果觉得我说话冒犯了你，那是因为我有点儿烦人，经常不认同别人的说法，所以有时要花很多时间论证，说服自己接受他人的观点。

本书结构

我将这本书分成了三个部分。

- 第一部分（第 1~7 章）：基础知识

开门见山，介绍如何使用 TDD 开发一个简单的 Web 应用。我们会先（用 Selenium）写一个功能测试，然后介绍 Django 的基础知识，包括模型、视图和模板。在每个阶段，我们都会编写严格的单元测试。除此之外，我还会向你引荐测试山羊。

- 第二部分（第 8~17 章）：Web 开发要素

介绍 Web 开发过程中一些棘手但不可避免的问题，并展示如何通过测试解决这些问题，包括静态文件、部署到生产环境、表单数据验证、数据库迁移和令人畏惧的 JavaScript。

- 第三部分（第 18~26 章）：高级话题

介绍模拟技术、集成第三方系统、测试固件、由外而内的 TDD 流程以及持续集成（Continuous Integration，CI）。

排版约定

本书使用了下列排版约定。

- 黑体

表示新术语或强调的内容。

- 等宽字体（Constant width）

表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。

- 加粗等宽字体（Constant width bold）

表示应该由用户输入的命令或其他文本。

偶尔使用 [...] 符号表示省略了一些内容，截断较长的输出，或者跳到相关的内容。



该图标表示提示或建议。



该图标表示提示、建议或一般注记。



该图标表示警告或警示。

提交勘误

发现了错误或错别字？本书的相关资源放在 GitHub 上，欢迎你随时提交工单和拉取请求：<https://github.com/hjwp/Book-TDD-Web-Dev-Python/>。

如果发现中文版有错误或错别字，欢迎提交勘误至 <http://www.ituring.com.cn/book/2052>。

使用代码示例

代码示例可到 <https://github.com/hjwp/book-example/> 下载，各章的代码都放在单独的分支中，请到 <http://www.ituring.com.cn/book/2052> “随书下载”处下载。附录 J 中有这个仓库的使用方法。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无须联系我们获得许可。比如，用本书的几个代码片段写一个程序就无须获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无须获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN。比如：“*Test-Driven Development with Python, 2nd edition*, by Harry Percival (O'Reilly). Copyright 2017 Harry Percival, 978-1-491-95870-4.”

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 permissions@oreilly.com 与我们联系。

O'Reilly Safari



Safari (以前叫 Safari Books Online, <http://www.safaribooksonline.com>) 是会员制平台, 为企业、政府、教学人员和个人提供培训和参考资料。

会员可以访问上千种图书、培训视频、学习路径、交互式教程和精心制定的播放列表。这些资源由 250 多家出版社提供, 包括 O'Reilly Media、Harvard Business Review、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Adobe、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett 和 Course Technology, 等等。

详情请访问 <http://oreilly.com/safari>。

联系我们

请把对本书的评价和问题发给出版社。

美国:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国:

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页, 你可以在那儿找到本书的相关信息, 包括勘误表、示例代码以及其他信息。本书的网站地址是:

<http://shop.oreilly.com/product/0636920029533.do>

对于本书的评论和技术性问题, 请发送电子邮件到: bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息, 请访问以下网站:

<http://www.oreilly.com>

我们在 Facebook 的地址如下: <http://facebook.com/oreilly>

请关注我们的 Twitter 动态: <http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下: <http://www.youtube.com/oreillymedia>

电子书

如需购买本书电子版，请扫描以下二维码。



准备工作和应具备的知识

我假设读者具备了如下的知识，电脑中还应该安装一些软件。

了解Python 3，会编程

写这本书时，我考虑到了初学者。但如果你刚接触编程，我假设你已经学习了 Python 基础知识。如果还没学，请阅读一份 Python 初学者教程，或者买一本入门书，比如 *Dive Into Python* 或《“笨办法”学 Python》，或者出于兴趣，看一下《Python 游戏编程快速上手》。这三本都是很好的入门书。

如果你是经验丰富的程序员，但刚接触 Python，阅读本书应该没问题。Python 简单易懂。

本书中我用的是 Python 3。我在 2013—2014 年写这本书时，Python 3 已经发布好几年了，全世界的开发者正处在一个拐点上，他们更倾向于选择使用 Python 3。可以参照本书内容在 Mac、Windows 和 Linux 中实践。在各种操作系统中安装 Python 的详细说明后文会介绍。



本书内容在 Python 3.6 中测试过。如果你使用的是较低版本，可能会发现细微的差别（比如 f 字符串句法），所以如果可以，最好升级 Python。

我不建议使用 Python 2，因为它和 Python 3 之间的区别太大。如果碰巧你的下一个项目使用的是 Python 2，仍然可以运用从本书中学到的知识。不过，当你得到的程序输出和本书不一样时，要花时间判断是因为用了 Python 2，还是因为你确实犯了错——这么做太浪费时间了。

如果你想使用 PythonAnywhere（这是我就职的 PaaS 创业公司），不愿在本地安装 Python，可以先快速阅读一遍附录 A。

无论如何，我希望你能使用 Python，知道如何从命令行启动 Python，也知道如何编辑和运行 Python 文件。再次提醒，如果你有任何疑问，看一下我前面推荐的三本书。



如果你已经安装了 Python 2，担心再安装 Python 3 会破坏之前的版本，那大可以放心，Python 3 和 Python 2 可以相安无事地共存于同一个系统中，使用虚拟环境的话（本书就是）更是如此。

HTML的工作方式

我还假定你基本了解 Web 的工作方式，知道什么是 HTML、什么是 POST 请求等。如果你对这些概念不熟悉，那么需要找一份 HTML 基础教程看一下，webplateform 网站上列出了一些。如果你知道如何在电脑中创建 HTML 页面，并在浏览器中查看，也知道表单以及它的工作方式，那么或许你就符合我的要求。

Django

本书使用 Django 框架，这（或许）是 Python 世界最为人认可的 Web 框架。本书不要求读者事先了解 Django，但如果你刚接触 Python、刚接触 Web 开发，也刚接触测试，偶尔会觉得话题太多，有太多的概念要理解。如果发生了这样的情况，我建议你先把本书放下，找份 Django 教程看看。DjangoGirls 是我所知的对新手最友好的教程。官方教程 (<https://docs.djangoproject.com/en/1.11/intro/tutorial01/>) 对有经验的程序员来说是不错的选择。

Django 的安装说明参见后文。

JavaScript

本书后半部分有少量 JavaScript。如果你不了解 JavaScript，先别担心。如果你觉得有些看不懂，我到时会的推荐一些参考资料给你。

关于 IDE

如果你来自 Java 或 .NET 领域，可能非常想使用 IDE（集成开发环境）编写 Python 代码。IDE 中有各种实用的工具，例如 VCS 集成。Python 领域也有一些很棒的 IDE。刚开始我也用过一个 IDE，它对我最初的几个项目很有用。

我能建议（只是建议）你**别用** IDE 吗？至少在阅读本书时别用。在 Python 领域，IDE 不是那么重要。写作本书时，我假定你只使用一个简单的文本编辑器和命令行。某些时候，它们是你所能使用的全部工具（例如在服务器中），所以刚开始时值得花时间学习使用基本的工具，理解它们是如何工作的。即便当你读完本书后决定继续使用 IDE 和其中的实用工具，这些基本的工具还是唾手可得。

需要安装的软件

除了 Python 之外，还要安装以下软件。

- Firefox Web 浏览器

Selenium 其实能驱动任意一款主流浏览器，不过以 Firefox 举例最简单，因为它跨平台。而且使用 Firefox 还有另外一个好处——和公司利益没有多少关联。

- Git 版本控制系统

Git 可在任何一个平台上使用。Windows 安装环境带有 Bash 命令行，这是本书所需的。

- 装有 Python 3、Django 1.11 和 Selenium 3 的虚拟环境

Python 3.4+ 现在自带 virtualenv 和 pip（早期版本没有，这是一大进步）。搭建虚拟环境的详细说明参见后文。

- Geckodriver

这是通过 Selenium 远程控制 Firefox 的驱动。在“安装 Firefox 和 Geckodriver”一节会给出下载链接。

针对 Windows 的说明

Windows 用户有时会觉得被开源世界忽略了，因为 macOS 和 Linux 太普遍了，很容易让人忘记在 Unix 之外还有一个世界。使用反斜线作为目录分隔符？盘符？这些是什么？不过，阅读本书时仍然可以在 Windows 中实践。下面是一些小提示。

1. 在 Windows 中安装 Git 时，一定要选择“Run Git and included Unix tools from the Windows command prompt”（在 Windows 命令提示符中运行 Git 和所含的 Unix 工具）。选择这个选项之后就能使用 Git Bash 了。把 Git Bash 作为主要命令提示符，你就能使用所有实用的 GNU 命令行工具，例如 `ls`、`touch` 和 `grep`，而且目录分隔符也使用斜线表示。
2. 在 Git 安装程序中，还要勾选“Use Windows' default console”（使用 Windows 的默认控制台），否则 Python 在 Git Bash 窗口中无法正常使用。
3. 安装 Python 3 时，除非已经安装了 Python 2 且想继续将它用作默认版本，否则一定要选中“Add Python 3.6 to PATH”（把 Python 3.6 添加到系统路径中，如图 P-1 所示），这样才能在命令行中顺利运行 Python。



测试所有软件是否正确安装的方法是，打开 Git Bash 命令提示符，在任意一个文件夹中执行命令 `python` 或 `pip`。