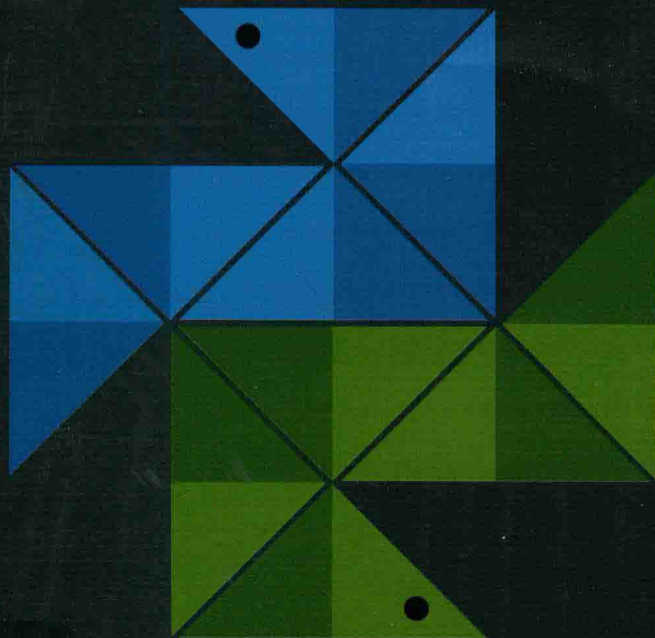



Python 编程基础 与 HTTP 接口测试

阿奎 编著

通天在线指导
高手疑难解答
资源·微课·视频



 中国工信出版集团

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

Python 编程基础 与 HTTP 接口测试

阿奎 编著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书主要包含四大部分内容，第一部分概括介绍了 Python 语言流行的原因，以及测试人员如何学习接口测试与自动化测试；第二部分为 Python 编程基础；第三部分是 HTTP 协议；第四部分是 HTTP 接口测试实战。前三部分让读者可以从零开始，逐步掌握 Python 语言，具备基本的 Python 编程基础。通过练习的方式了解 HTTP 协议中常用的概念和机制。最后，第四部分将两者相结合，使读者学会用 Python 进行 HTTP 接口测试，主要采用 Python 的单元测试方法进行 HTTP 接口测试工作。通过一步一步的练习，读者可掌握 Unittest 单元测试框架的使用，掌握测试数据的外部化（到文件和到 Excel 等），掌握 HTTP 接口的发现和探测。

本书亦学亦练，学练结合，每个章节分为多个主题小节。小节的前半部分会对本小节的知识主题进行详细的介绍，后半部分会启动一个练习，让读者边阅读边练习，在练习中检验学习的成果。

本书适合所有对 Python 语言和测试感兴趣的程序员、编程者、测试人员，也适合高校计算机专业学生补充学习、扩充视野。

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

Python 编程基础与 HTTP 接口测试 / 阿奎编著. —北京：电子工业出版社，2018.1
ISBN 978-7-121-32995-1

I. ①P… II. ①阿… III. ①软件工具—程序设计②计算机网络—通信协议 IV. ①TP311.56②TN915.04

中国版本图书馆 CIP 数据核字（2017）第 264345 号

策划编辑：张瑞喜

责任编辑：张瑞喜

印 刷：中国电影出版社印刷厂

装 订：中国电影出版社印刷厂

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：710×1000 1/16 印张：14.5 字数：252 千字

版 次：2018 年 1 月第 1 版

印 次：2018 年 1 月第 1 次印刷

定 价：45.00 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）88254888，88258888。

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

本书咨询联系方式：zhangruixi@phei.com.cn。

序 1

身处软件开发这个颇具颠覆性的行业，又恰逢智能技术的爆发期，很多认知都在被刷新着。和阿奎相识源于组织的敏捷转型合作，初几次见面印象是此人技术问得很细，聊了很多关于代码重构和自动化测试的观点，这可能也是敏捷圈子里最能够试探彼此是否为圈内人士的方法。那个时候关于“测试已死”的观点还是颇为流行的，当然意思是传统的人肉测试最终都会被自动化测试所取代，而开发人员通过测试驱动开发这样的技术实践自然就能生成自动化测试集，那专职测试还留着作甚？

前两周圈子里的人工智能专家给我转来一篇论文，快速扫描后大意是一组科研人员正通过语义和图形分析的技术对算法类论文进行直接算法生成，也就是说给这程序一个算法描述，它就还你一个真正的算法实现。当然现实中给一个高保真网页图片，还你一个真实网站的在线服务已经存在了。稍加思索你会得出一个具有讽刺意味的推论：是不是“开发已死”？具备一定智能的程序已经可以自己产生相应的程序了。而测试却好像并没有死，反而这两年从探索性测试到针对新交互渠道的感知测试还越来越火了。

看到这本书的时候，很高兴当年和阿奎讨论的一个观点现在仍然是对的，那就是“测试人员必须写代码”。不管是现代的分布式互联网应用，还是即将到来的物联网设备，其测试的难度都提升了很多，甚至可以说如果不采用自动化的程序是不可能进行有效测试的。想像一下把应用商城刷积分的人肉模式复制到未来一平米可能有上万个物联网设备接入的场景下，你可能需要一个 Alpha Go 来完成这样的“人肉”测试。有幸的是我认识了几位把测试人员写代码真正贯彻和执行下去的管理者，阿奎就是其中一位。

Python 作为编程语言自不用多讲，曾经在测试领域因为其灵活性大受欢迎，积累了不少有用的测试库，即使现在很多项目上我们还会日常使用。而大数据



时代的来到更催生了这门语言的学习热潮，不少业务人员开始学习 Python 作为数据分析的编程语言。时下的大数据分析不同于之前既定规则的数据处理，更多是在大数据中去挖掘和探索，某种意义上我们又何尝不可以说是在进行“数据测试”呢？所以学习 Python 应该是符合时代背景高投入产出比的事情。

从互联网到移动互联网，再到物联网，很多技术都会被改变，最有可能持续发光发热的“组件”可能就是我们的应用通信协议 HTTP 了。针对 HTTP 通信的接口测试也是最常见的，经常也是最繁琐的。学习如何进行有效的自动化是避免自己被机器人取代的必备技能。

最后，很感谢阿奎把自己的学习心得和方法也融汇到了此书中。刻意练习是技术学习的不二窍门，如何坚持确是相当有难度的一件事情。和广大读者一样，阿奎并非身处一个纯粹的技术公司，日常有很多的管理会议和业务需求，所以他的学习经验分享更值得大家借鉴和试验。

肖然

精益敏捷专家

ThoughtWorks 咨询与设计总监

序 2

转眼，阿奎工作已经 13 个年头了，先后做过程序员、软件工程师、团队经理，现在是一名系统分析师和内部敏捷教练，积极地参与组织中的敏捷实践和推广工作。

在担任团队经理期间，阿奎先后带过 C/C++、Cobol 和 Java 为开发语言的开发团队，多年来一直保持对技术研发工作的热爱。最有意思的是，多年以前他还在一个以主机 Cobol 为开发语言的部门的时候，竟然会去参加以 Java 为语言栈的编程大赛。

接触最多的是 2013 年调到新技术实验室的时候，那个时候互联网金融风头正盛，敏捷开发方法作为一种新的软件开发方法，正在被各个大型企业所认可和尝试。那是他接触敏捷的缘起，甚至可以说也是这本书的缘起。

阿奎自 2013 年开始从事敏捷实践和推广工作，他将自己定位为一名技术性管理者，在从事团队管理的同时，一直保持着对各种新技术的关注，在工作中对于如何提升团队的工作效率也多有思考。众所周知，敏捷开发转变的核心在于人的转变，而在人的转变过程中，人员能力的提升和对新的开发模式的适应又是尤为重要的。

随着敏捷、精益、DevOps 等新的软件开发理念的盛行，软件测试的从业人员也正在经历工作环境和工作要求的改变。这种改变需要软件测试从业者成为真正的 IT 人，具备计算机基础理论知识和网络知识，掌握一门编程语言，熟悉一个操作系统和一个数据库。

环顾业内，很多软件测试从业者都正在经历从“点点点”工程师到自动化测试工程师的转变，这个转变是痛苦的，也是迷茫的。阿奎就是看到了这样一个需要，利用一年的业余时间，写了这本书，其中的每一个练习都是他精心设计的，并且都自己进行了实战和解答。这本书是他专门针对没有编程基础的软



件测试工程师如何快速、高效地学会编程这一课题的一次实践和有益的尝试。

书中提到的“闯关式学习”实际上就是“刻意练习”的学习思想，在编程语言学习上的实地应用，并且经过他的一些实验，取得了非常好的学习效果。

特别祝贺阿奎，能够将自己长期工作、学习的收获和心得，通过一本书的形式进行一次总结，特别是这样一本帮助大家学习编程的书。

薛勇

主任工程师

中国银行软件中心

前言

每个人都应该花 1 年时间学习编程。

——史蒂夫·乔布斯

我一直以来有一个观点，对于希望从事自动化测试工作或者希望掌握自动化测试技能的人士，掌握一门计算机语言是绕不过去的一个坎。就是这个坎，让很多从事手工测试的工程师对于自动化测试工作望而却步，也让很多人不由自主地退而求其次，转向去学习一些自动化测试的工具，来回避语言学习的困难。

看到身边很多的测试工程师对待编程语言的学习畏之如虎，谈之无力，让我产生了“为希望转型成为自动化测试工程师的软件测试从业者提供一本靠谱的编程语言学习指导书”的想法。这也是促成这本书初稿形成的原始动力。

随着信息技术的发展，计算机办公技能成为一名现代白领工作者的必备技能，我们日常工作中有很多的时间都是在与计算机进行交互。通过对计算机语言的学习，可以为你打开一扇与计算机进行深层次互动交流的大门。同时，在编程语言的学习中，你会学到一种新的思考方式并找到另一种看待问题、解决问题的视角。

本书的内容包含两部分：Python 编程基础和基于 Python 的 HTTP 接口测试。

在 Python 编程基础部分，主要关注 Python 语言的基础知识的学习和掌握，这一部分对于每一名希望快速掌握一门计算机语言的学习者都是适用的。

基于 Python 的 HTTP 接口测试部分，重点面向希望转型为自动化测试工程师的软件测试从业者，以 HTTP 接口测试为应用场景，来学习和掌握 Python 的相关知识点和使用技巧。

本书以闯关式学习方法为指导进行编写，让读者通过一个一个小的关卡的

刻意练习，在不知不觉中掌握“Python 编程基础”和“HTTP 接口测试”的相关技能，完成从手工测试工程师到自动化测试工程师的转变。

所谓闯关式学习方法，通俗地讲，就是通过不断的刻意练习，打通一个一个的练习关卡来进行自我提升和学习的方法。闯关式学习方法首先要求有一名有经验的指导者，针对学习目标刻意设计的一个又一个的练习关卡，即挑战问题，读者通过一定的前期知识的学习和练习后，在给定的时间内对第一个关卡的问题发起挑战，即闯关。如果给定的时间内不能有效地完成挑战问题，就需要反复的刻意练习，直至轻松完成挑战问题为止，即闯关成功。此时，才可以进入下一个练习关卡。详细的关于如何使用本书进行闯关式学习的内容，请读者阅读本书的“如何阅读和使用本书”部分。

我一直认为，学习软件开发、测试技能，和学习骑自行车、游泳一样，是一项技能的修炼，而非仅仅是知识或者概念的了解和掌握。要习得一项技能需要刻意地练习。

编程作为一门技艺，是可以习得的，习得是有方法的！这本书为愿意学习的读者提供了习得编程技艺的方法，就是“闯关式学习”。

但是，“没有任何有效的学习和精进的过程是惬意的和不需要付出努力的。”

所以，这是一本针对 Python 编程基础和 HTTP 接口测试技能，进行刻意练习的学习指南，这不是一本可以靠在沙发上阅读的消遣书。

“学习之路挖坑容易挖井难”，祝大家利用这本精心打造的“闯关之书”，挖出属于自己的“Python 之井”。

阿奎（于洪奎）

如何阅读和使用本书

相信很多读者都有过自学软件测试或者其他技能的经历，期间坚持与放弃的挣扎，学习方向的迷茫，学习资料的收集，学习内容选择……甘苦自知。

作家格拉德威尔在《异类》一书中指出：“人们眼中的天才之所以卓越非凡，并非天资超人一等，而是付出了持续不断的努力。1万小时的锤炼是任何人从平凡变成超凡的必要条件。”他将此称为“一万小时定律”。要成为某个领域的专家，需要10000小时，按比例计算就是：如果每天工作八个小时，一周工作五天，那么成为一个领域的专家至少需要五年。

我并不是想用这段引用来告诉大家，成为自动化测试工程师需要五年的时间，而是希望大家看到其中“锤炼”一词的深意——成就超凡技艺的过程，实际上是一个“锤炼”的过程，也就是“刻意练习”的过程。

既然叫做刻意练习，有别于随意的练习。随意的练习并不能带来有效的学习和精进，并且会浪费时间，打击练习者的信心。没有任何有效的学习和精进的过程是惬意的，不需要付出努力的，要做到刻意练习有如下四个要点：

- 有目的的练习。刻意练习一定是针对某一个既定的目的进行练习，即每一个练习都是有具体目标的。
- 精神高度集中的练习。刻意练习一定是需要练习者精神高度集中，全身心投入的，只有这样才能带来学习上的领悟和突破。
- 反复的练习。在以上两点的基础上，重复的练习是技艺提升不可缺少的一环，没有什么技巧性的技艺是一次就可以完全掌握和彻底领悟的，编程尤其如此。只有通过不断的重复的练习才能将编程中用到的一些基本的概念和模式，变成自己的肌肉记忆和大脑反射。
- 获得有效反馈的练习。反复练习中，还需要有效的反馈来确定练习是否取得了进展，这种反馈一方面可以促进练习者更好的坚持练习，另一方

面也可以告诉练习者，练习到什么程度就够了。

以上也是闯关式学习方法的理论依据和本书设计编写的初衷。

本书采用“闯关式学习方法”编写。阅读的过程中，需要读者边读边练习，每一章包含多个小节，每个小节都是一个小的关卡。

再次重申：这是一本针对 Python 编程基础和接口测试技能进行刻意练习的学习指南，这不是一本可以靠在沙发上阅读的消遣书。

读者在使用这本书的时候，应该是在电脑旁边的，并且电脑是联网的，这样你才能通过刻意练习，成功挑战书中一个一个的小节关卡，体会成功闯关后的喜悦和掌握一个一个技能要点后发自内心的欢喜。

小节关卡

本书的每一个小节一般包含：学习目标、学习资源、知识准备、挑战问题、难点提示、知识总结、拓展问题七个段落。

学习目标：描述本小节的学习目标，即通过本小节的学习掌握的知识要点。

学习资源：当今社会已经是一个信息资源极其丰富的社会，我们不再缺乏信息，而是缺乏优质的，经过整理的高质量信息，本小节会提供一些与学习目标相关的高质量网络学习资源的链接，方便读者进行扩展阅读和深入学习。

知识准备：本小节主要就完成学习目标，需要学习和掌握的概念进行阐释。本小节的阐释不求全面，主要以常用的知识和概念以及技能的阐释和演示为主。更全面详细的信息，读者可以通过学习资源小节中提供的链接进行深入的学习。

挑战问题：本小节要解决的问题。注意只有解决问题才算通过本小节的学习，如果上一个小节的问题没有解决，不建议继续学习，因为，后面小节的解题和闯关过程中一般会用到前面小节的知识点和技能。同时，也是最重要的，小节问题的解决过程中可以借助互联网或者其他资料，但是，编码过程中，必须保证完全自主完成，也就是，在开始解题时候，应该自己完全手工完成，不能直接拷贝粘贴，也不要对着屏幕或者书本看一个单词敲一个单词。软件自动化测试的学习从归类上属于技能类学习，而技能类学习的关键就是刻意练习。所以，闯过关卡不是目的，重要的是切实掌握闯过关卡用到的知识点和技能。

难点提示：在进行小节闯关的过程中可能会碰到一些陷阱或者难于解决的



问题，会在这一部分给予一些提示。

知识总结：将本小节的知识点进行简单列举，需要读者根据知识点中的提示，翻看提供学习材料，对知识点进行深入的学习和必要的记忆、理解。

拓展问题：有的关卡的题目会有一些变化的问法或者多个解答方法，会在这一部分给出说明和提示。

使用指南

(1) 学习顺序：建议首先通读一遍本小节的所有内容，再通过翻阅提供的学习资料将知识点中提到的内容记忆和理解，最后再进行本小节问题的解决，即实施闯关，闯关成功后，再将知识总结小节的内容做一个回顾，然后再开始下一个小节的学习。

何为“闯关成功”？即在给定时间内独立完成挑战题目。在完成挑战题目的基础上，有两个要求：在给定时间内独立地完成。

如果完成挑战题目的时间过长或者过程中有一边看资料一边写代码的情况，这都不能算作闯关成功。此时，建议读者删除代码重新进行闯关。

(2) 刻意练习：

- 不要拷贝粘贴。
- 不照抄照搬。
- 十分钟原则。
- 举一反三，主动拓展。

备注：

“十分钟原则”：每一个小节的练习问题需要在十分钟内解决，一般前面几个小节，读者都不会有问题，但在后面的时候，便很难在这个时间里完成。而所谓的“十分钟原则”就是，如果你没有在十分钟的时间里完成小节的练习，就将代码删除，重新写一遍，是在编程学习中对“刻意练习”的贯彻。

“举一反三，主动拓展”又称为“不以闯关为目的的闯关”。有的同学练习过程中对于闯关很有兴趣，又有一点编程基础，一上来就一口气闯到了第五关，甚至第九关，这样的闯关是没有意义的。这些关卡的设计过程中，笔者针对每一个关卡都做了两次以上的练习，就是至少删除过一次代码。目的就是更好地

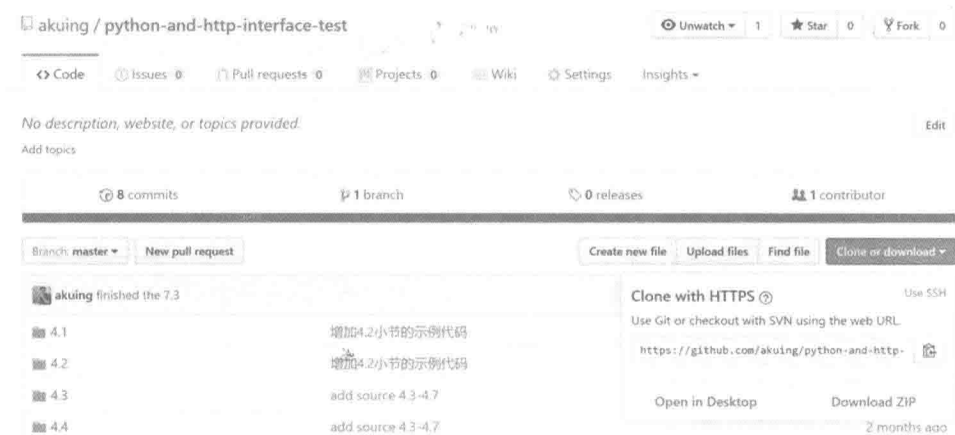
体会学习目标和知识点，更好地形成所谓“肌肉记忆（muscle memory）”。经过思考，笔者还在多个章节设计了拓展环节，就是为了更好地掌握相关知识和技能。“不以闯关为目的的闯关”，才是这些关卡设计的本意。

（3）找到组织：

学习过程中会遇到很多的问题，如果有老师或者同行者，一起互相激励，遇到问题也可以互相请教，在交流中收获更多。读者可以关注本书的公众号：IT 匠艺教研室。公众号中有微信交流群，可以扫描加入，随时进行交流和學習。

为方便读者自学，笔者将本书各个小节的练习代码上传到了 GitHub，读者可以自行下载参考。

地址为：<https://github.com/akuing/python-and-http-interface-test>。点击右侧的绿色按钮“clone or download”，选择 Download ZIP 即可以下载整个参考代码和示例的压缩包。



本书编写的目的是通过练习，让读者将技能真正掌握到自己手中。所以，不要为了闯关而直接下载运行参考示例代码，这对你的学习不会有任何实际的帮助，只会让你浪费掉一个作者苦心设计的用于检验学习进展的关卡。

所以，建议仅仅在自己已经完成了关卡的要求之后，再参考这些示例代码，与其互相取长补短，巩固所学。

再次强调，这不是一本只是用来“看”的书，这是一本练习的路线图，如

果学习过程中有任何困难，可以发邮件或者加入公众号：IT 匠艺教研室，进入公众号中的学习交流群询问，我非常愿意与学习者交流学习的问题，以帮助更多的读者闯过学习第一门语言的大山。

学习纪律

实施闯关的过程要遵守如下学习纪律：

- 按部就班不跳关。
- 自主学习不拷贝。
- 刻意练习不照抄。
- 新词新知要记牢。

以上学习纪律，前三条就不多说了，对于“新词新知要记牢”进行一下特别的阐述。学习 Python 的过程中会出现很多新的知识点和新的词汇，其中不少是英文的。这对于英文基础不好的学习者会是一个很大的障碍。作者在十几年的工作中也曾看到很多学习者都是因为英语这一关没有过，导致编程语言的学习进步缓慢。

翻开这本书时，每一位读者的英语水平已经是一个客观事实，并不会因为读到了本书而有什么改变，并且，即使学习完这本书，学习者的英语水平仍然不会有什么改变。作者这里希望针对编程语言的学习和英语水平之间的关系做一个阐述。

同样的学习条件，英语基础好的，学习编程语言的时候会更加容易一些。

新知识一般有两种方式被大脑接纳。第一种是挂接或者建立在已有的知识体系上；第二种就是在大脑里重新打桩，建立记忆根。这两种模式下，对于同样的知识点的掌握，所要付出的努力是完全不同的，而这种不同主要来自不同个体大脑中已有的知识体系的差异。

比如，对于已经有一定英语基础的同学，在进行 MySQL 的学习时就会很容易记住 create、delete、update、select 这些 SQL 的关键字以及他们的用途；而对于一个英语基础薄弱的同学，如果他之前并不认识这四个单词，甚至只认识 26 个字母（不认识 26 个字母的读者，本文帮不到你太多，建议不要再浪费自己的时间看下去了，这里没有任何偏见和轻视，仅仅是真诚的提醒！），要记住这四个关键字，将不是一件容易的事情。为什么会这样，其实很简单。对于熟悉这四个单词的人，由于已经拥有了对单词自身含义的理解，他看到这四个单词的时候看到的是四个整体，他所要记忆的只是四个元素，至于每一个元素

的拼写、意义区分已经是他已有的知识体系里的内容；而对于不认识这四个单词的人，他要记忆的是 24 个元素，并且每六个元素被作为了一组，拥有一个固定的含义和拼写顺序。虽然从知识点来说两个人都是要掌握 MySQL 的四个 SQL 关键字，知识点是相同的，但是对于两个不同的个体，这个知识点的信息量是完全不同的。

这就是为什么“同样的学习条件，英语基础好的，学习编程语言的时候会更加容易一些。”的原因。

英语基础差的学习者，也是可以学会编程语言的，只是需要付出更多的努力，并掌握一定的方法。

对于英语基础薄弱的学习者，我有如下两点建议：

(1) 从意识上认识清楚，自己就是要比英语好的同学付出更多的努力才能学好编程语言。因为，这个已经是既定的客观事实，并不会因为我们无视它，不去面对它而消失。并且，现实中很多英语基础并不好的同学的确通过自己的努力学会了一门编程语言，我们也没有必要非要要求自己一定要学好英语之后，才去学习编程。

(2) 从方法上区别对待，对于编程语言学习过程中遇到的自己不认识的英语单词，按照英语的学习方法进行学习和记忆。具体来说，对于英语基础薄弱的学习者，我建议准备一个英语单词本，遇到自己不认识的英语单词，查字典并进行整理、记忆和默写。

总之，编程是一门技艺，不同的学习者会有不同的学习道路，而“刻意练习”是掌握这门技艺的不二法门。我真心的希望可以通过这本小书，让更多人，特别是那些希望转型成为自动化测试工程师的软件测试从业者，克服对编程语言的恐惧，真正的掌握这门即有用又好用的编程语言。

在线学习与疑难解答

本书特别为广大读者精心准备了专有的在线服务与交流平台。用微信扫描右侧二维码，关注“悦读力”，即可加入本书读者圈，获得关于 Python 编程及自动化测试相关的更多学习资源，并有机会向阿奎老师提问、得到在线学习指导。



读者圈

感谢

写书之前，我真的没有想到写一本书要付出这么多的精力。真的要特别的感谢在这本书的写作过程中，帮助和支持我的人。

有太多的人需要感谢，那么就让我按照时间的先后一一谢过吧！

本书的编写完全是出于对我日常工作中的所听所见的思考，我的身边有很多非常喜欢学习新技术的同事，他们有的从事开发、有的从事测试，有的甚至是管理人员——我为什么要用“甚至”——？在这本书还是一个电子的试读版本的时候，我就组织了一个利用业余时间的学习小组（测试融合训练营），主要是希望看看这样一本书是否真的能够帮到大家，没想到一经提议，大家就非常积极地参与，非常感谢我的同事白雪、史丽珍、关晓康、毛雪涛、张培、刘会琴、于国双(ygs)、农倩倩、胡江海，他们是电子版初稿的第一批阅读用户，在这一过程中大家给了很多有益的反馈。



xuer



史丽珍



Dracu...



阿奎



茅雪涛



张培



刘会芹



ygs



茜茜



上海...



群聊名称

测试融合训练营

同步的还有一个来自互联网的学习小组（用 Python 做 HTTP 接口测试学习班），主要是觉得同事们给建议的时候都会比较“委婉”，希望有一个来自互联网的测试和反馈。感谢杨艳艳、高园园、CJoy、王芳、汤濮瑜、海底小鱼的热情参与和学习过程中的积极反馈。特别是高园园在参与之后，写了一篇非常值得称赞的学习感想。这篇文章，我在征得她本人同意之后，附在了本书的附录中供大家参考。

在电子版的书经过验证之后，我就有了将其变成纸质书的想法，机缘巧合的是，在参加一次敏捷技术交流的时候，遇到了张瑞喜老师，在张老师的鼓励下，我才真正开始了这本书的编写过程，编写过程中，非常感谢张老师对一个初次写书的作者的耐心和辅导，特别是这本书的题目《Python 编程基础与 HTTP 接口测试》就是张老师取的，本书的电子版在“百度阅读”上的名字叫做《用 Python 做 HTTP 接口测试》。

这本书的第一个审校的人是我的太太悠悠女士，感谢她指出我书中原来的 28 处没有原因的“所以”。其实，我知道更应该感谢的，是她背后默默的支持，长时间的写作，占用了很多本应该陪她的时间。