Broadview
www.broadview.com.cn

人生苦短，我用Python
零基础起步，手把手进阶，Python这样学才简单！

跟老齐学 Python 轻松入门

齐伟 编著

本书是老齐专为“零基础”朋友学习Python而作，
其电子版已经经过无数在线网友验证。
本书吸取了很多网友提出的建议，在电子版基础上进行全新修订，更易懂，更系统，更合理。

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

跟老齐学 Python 轻松入门

齐伟 编著

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

内 容 简 介

Python 是一种被广泛用于网站开发、数据处理和机器学习等领域的高级编程语言，同时也是一种学习门槛较低的高级编程语言。本书是 Python 语言的入门读物，旨在引导初学者能够在轻松的环境中掌握 Python 的基础知识，包括基本对象类型、函数、类、模块以及数据存储方式。

本书适合计算机高级编程语言零基础水平及其以上的 Python 初学者阅读。

图书在版编目（CIP）数据

跟老齐学 Python：轻松入门 / 齐伟编著. —北京：电子工业出版社，2017.4
ISBN 978-7-121-30662-4

Ⅰ. ①跟… Ⅱ. ①齐… 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（2016）第 308414 号

责任编辑：高洪霞
印　　刷：北京京科印刷有限公司
装　　订：北京京科印刷有限公司
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：787×1092　1/16　印张：20.75　字数：541.2 千字
版　　次：2017 年 4 月第 1 版
印　　次：2017 年 4 月第 1 次印刷
定　　价：59.00 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 zlts@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：（010）51260888-819，faq@phei.com.cn。

自序

我曾经在网上写过《零基础学 Python（第 1 版）》，完成之后，发现有一些错误，并且整体结构对零基础的学习者来说还不是很适合。于是，就重新写了，后来有幸得到了电子工业出版社的认可，集结成为《跟老齐学 Python：从入门到精通》一书出版。但是，当书印出来之后，我发现还得修改，于是在原来的基础上又进行了修订，并且定名为现在的书名——《跟老齐学 Python：轻松入门》，言外之意，还有入门之后的教程。

本书也可以说是对已经出版的《跟老齐学 Python：从入门到精通》一书的修订和删减。原来那本书宣称“精通”，但很难做到精通。所以，这次修订就索性专注于入门。

首先，这是一本面向“零基础”学习 Python 语言的书，不是写给中高级程序员的。目的在于帮助“零基础”的读者入门。它不会让你在看完之后就达到精通 Python 的水平，但是它能够让读者窥视到 Python 语言的魅力，能够理解用 Python 编程的基本思想，搞明白 Python 的基础知识，从而为后续的“精通”奠定基础。

其次，本书也不是那种“*n* 个实例精通”的书。因为在我看来，通过简单几个例子就掌握一种语言，或许可以，但是不符合我的认知。如果读者喜欢“*n* 个实例精通掌握”，请移步到别处。

然后，本书还是一本比较有“水分”的书。很多读者希望有干货，Python 文档是最典型的干货，如果执意要求干货的朋友，请直接去看文档。本书中的“水分”是一种最好的溶剂和调味品，能够让你在阅读的时候不至于感到乏味。

当本书完成后，我还一直心怀惶恐，唯恐因为本书中的不当阐述而耽误了读者的前程。所以，建议读者在阅读本书的时候，如有怀疑，请更多地求助于搜索引擎（推荐使用 Google）或者其他资料，让自己对相应内容有更深入、全面、正确的理解。

欢迎读者提出意见或建议，以帮助我改进本书。所以，提供如下可以联系到我的途径：

（1）加入 QQ 群，可以跟很多人交流。QQ 群：Code Craft，26913719。

（2）关注我的新浪微博，名称是：老齐 Py。地址：http://weibo.com/qiwsir。

（3）到 github.com 上直接 follow 我，名称是：qiwsir。地址：https://github.com/qiwsir。

（4）经常关注我的网站：www.itdiffer.com，这里不仅为读者提供了本书的源码，而且还有其他深入学习的内容。

在本书的编写过程中，家母住院，我不得不在病榻旁完成了本书的部分内容。在医院里，看到的常常跟外面不同，也颇感生命的珍贵。所以，“人生苦短，请用 Python”不是简单的调侃。中秋将至，母亲已经无恙，愿天下的母亲和父亲都身体健康。阅读本书的读者，在你忙碌的学习和工作之余，要挤出时间陪伴父母——有时候觉得是煽情的话，在经历之后发现绝非如此。

最后需要说明，本书虽然再次修订，但也难免有错误和不当之处，敬请读者指出。

齐伟

目录

第0章

预备

因为本书的读者是零基础学习 Python 的朋友，所以这里单独设置了一个预备章节，向开始阅读本书的读者介绍有关 Python 的发展情况，以及如何才能拥有一个可以进行 Python 程序开发的环境。

0.1 关于 Python 的故事

不管是学习某种自然语言（如英语），还是学习某种编程语言（如汇编），总要说一说有关这种语言的故事。

0.1.1 Python 的昨天、今天和明天

这个题目似乎有点大了，回顾过去、考察现在、张望未来，都是那些掌握方向的大人物做的事。那么现在就让我们每个人都成为大人物吧，因为如果不回顾一下历史，就无法满足学习者的好奇心；如果不考察一下现在，学习者就会担心学了之后没有什么用途；如果不张望一下未来，那么又怎么能吸引学习者或未来的开发者呢？

1. Python 的历史

Python 的创始人为吉多·范罗苏姆（Guido van Rossum）。关于他开发这种语言的过程，很多资料里面都要记录下面的故事：

1989 年的圣诞节期间，吉多·范罗苏姆为了在阿姆斯特丹打发时间，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。之所以选中 Python 作为程序的名字，是因为他是一个蒙提·派森的飞行马戏团的爱好者。ABC 是由吉多参加设计的一种教学语言。就吉多本人看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有成功，究其原因，吉多认为是非开放造成的。吉多决心在 Python 中避免这一错误，并取得了非常好的效果，完美结合了 C 和其他一些语言。

这个故事是笔者从《维基百科》里面直接复制过来的，很多讲 Python 历史的资料里面也都

转载过这段。但是，在笔者来看，这段故事有点忽悠人的味道。其实，上面这段中提到的，吉多为了打发时间而决定开发 Python 的说法，来自他自己的这样一段自述：

Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office (a government-run research lab in Amsterdam) would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project,being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).（原文地址：https://www.python.org/doc/essays/ foreword/）

首先，必须承认，吉多是一个牛人，非常牛的人。

其次，读者千万别认为 Python 就是随随便便、牛人一冲动搞出来的东西。牛人也是站在巨人的肩膀上的。

最后，牛人在成功之后，往往把奋斗的过程描绘得比较简单，或者是因为谦虚，或者是为了让人听起来更牛，反正，我们看最后结果的时候，很难感受过程中的酸甜苦辣。

不管怎么样，牛人在那时候发明了 Python，而且，他更牛的地方在于具有现代化的思维开放。通过 Python 社区，吸引来自世界各地的开发者参与 Python 的建设。在这里，请读者一定要联想到 Linux 和它的创始人芬兰人林纳斯·托瓦兹。两者都秉承“开放”思想，得到了来自世界各地开发者和应用者的欢迎和尊重。

2．Python 的现在

应该说，Python 现在表现不错。除了在 Web 开发方面有很多应用之外（当然 PHP 在这方面也很不错），在数据分析、机器学习、大数据、云计算等这些时髦的领域也都有它的身影，并且影响力越来越大。此外，还有自动化运维、自动化测试等。

读者可以到这个网站看一看 Python 的应用案例：https://www.python.org/about/success/。

不过，因为大学教育的问题，致使很多青年才俊对 Python 了解甚少；更因为学以致用的功利传统，青年才俊们最担心的是学了 Python——这种学校老师很少甚至从没有提及的怪东西没有什么用途，因为青年才俊们已经被铺天盖地的“学开发，做 APP，30 岁之前实现财务自由”的广告所包围，误以为“软件开发=做 APP”，其他都过时了。希望青年才俊们能够跳出四角天空，用自己的头脑思考问题、用自己的眼睛看世界，形成独立的判断，不要听信广告，也包括笔者在这里对 Python 的各种溢美之词。

3．Python 的未来

这个不需要描述，Python 的未来在所有使用者和学习者手中。

而且，从当前的发展来看，Python 的未来还是相当光明的。在软件开发领域，能不能说美利坚合众国的今天就是我们的明天呢？如果能，那么学 Python 就绝对不会吃亏。

0.1.2 优雅的 Python

Python 号称是优雅的。

这是一种仁者见仁、智者见智的说法。比如经常听到大师们说“数学美”，是不是谁都能体验到呢？不见得。

所以，是不是优雅、是不是简单、是不是明确，只有“谁用谁知道”，只有内行人才能理解。

不过，笔者特别喜欢下面这句话：**人生苦短，我用 Python**。

Python 能够提高开发效率，让你短暂的人生除工作外，还有更多的时间休息、娱乐或者做其他的事。

或许有的人不相信，那就比较一下吧。

0.1.3 与其他语言比较

“如果你遇到的问题无法用 Python 解决，那么这个问题也不能用其他语言解决。”——这是笔者向一些徘徊在 Python 之外的人常说的，可能有点夸张了。

有一篇题为《如果编程语言是女人》（网址：http://www.vaikan.com/if-programming-languages-are-woman/）的文章，笔者引用其中的部分内容作为不同语言的比较：

PHP 是你豆蔻年华时的心上人，她是情窦初开的你今年夏天傻乎乎追求的目标。玩一玩可以，但千万不要投入过深，因为这个“女孩”有严重的问题。

Ruby 是脚本家族中一个非常漂亮的孩子。第一眼看她，你的心魄就会被她的美丽摄走。她还很有趣。起初她看起来有点慢，不怎么稳定，但近些年来她已经成熟了很多。

Python 是 Ruby 的一个更懂事的姐姐。她优雅、新潮、成熟。她也许太过优秀，以致于很多人喜欢她。你把她当成了一个脾气和浪漫都退烧了的 Ruby。

Java 是一个事业成功的女人。很多在她手下做过事的人都感觉她的能力跟她的地位并不般配，她更多的是通过技巧打动了中层管理人员。你也许会认为她是一个很有智慧的人，并且愿意跟随她，但你要做好在数年里不断地听到“你用错了接口，你遗漏了一个分号”这样的责备的准备。

C++ 是 Java 的表姐。她在很多地方跟 Java 类似，不同的是，她成长于一个天真的年代，不认为需要使用“保护措施”。当然，“保护措施”是指自动内存管理。

C 是C++的妈妈。对一些头发花白的老程序员说起这个名称，会让他们眼前一亮，产生无限回忆。

Objective C 是 C 语言家族的另外一个成员。她加入了一个奇怪的教会，不愿意和任何教会之外的人约会。

以上只是娱乐，或许存在争议，权当参考吧。

严肃地说，Python 值得拥有。

在正式开始学习 Python 之前，首先要告诉大家 Python 的要诀。

0.1.4 《Python 之禅》

《Python 之禅》（The Zen of Python）包含了 Python 的特点说明和使用方法，当然，第一遍读到它可能没有什么感觉，但当你阅读完本书之后再读一读它，会对其中的每句话都有不一样

的理解。它就是 Python 秘籍。

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than right now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

“吃水不忘挖井人”，对于创造了 Python 的人，我们一定要感恩并崇拜。

0.1.5 感谢 Guido van Rossum

Guido van Rossum 是值得所有 Pythoner 感谢和尊重的，因为他发明了这个优雅的编程语言。他发明 Python 的过程是那么的让人称赞和惊叹，显示出牛人的风采。

Python 已经让人心动了。除了心动之外，还要行动；只有行动，才能“从小工到专家”。

0.2 从小工到专家

这是每个程序员的梦想。

有一本书的名字就是《程序员修炼之道：从小工到专家》，在这里向读者推荐此书，并借用该书标题。

本书或许能够成为你专家路上的一块铺路石，如果真能如此，笔者感到荣幸之至。祝所有读者都能成为专家。

0.2.1　Python 的版本

关于 Python 的版本问题，是必须要交代的。

不管出于什么原因，笔者认为 Python 给自己搞了两个版本，是败笔，但为了对某些比较底层的东西进行修改，这个败笔也是无奈之举，是值得的。

虽然如此，但幸亏两个版本并非天壤之别，绝大部分是一样的。所以，学习者可以选择任何一个版本进行学习，然后在具体应用的时候，用到什么版本，只要稍加注意，或者到网上搜索一下即可。

推荐阅读一篇参考文章《Python 2.7.x 和 3.x 版本的重要区别》（https://github.com/qiwsir/StarterLearningPython/blob/master/n005.md），供读者了解这两个版本之间的差异。

但是，总有不放心的初学者。

笔者曾被无数次地拷问：是学习 Python 2 还是 Python 3？

以前的版本是 Python 2，但是，总要与时俱进，本书则为 **Python 3**。

不管是 Python 2 还是 Python 3，总要从零开始学习，这意味着不需要基础。

0.2.2　学习 Python 是否需要基础

这是很多初学者都会问的一个问题。诚然，在计算机方面的基础越好，对学习任何一门新的编程语言越有利。如果你在编程语言的学习上属于零基础，那么也不用担心，不管用哪门语言作为学习编程的入门语言，总要有一个开始。

就笔者个人来看，Python 作为学习编程的入门语言是非常适合的。凡是在大学计算机专业学习过 C 语言的同学，都会体会到 C 语言不是很好的入门语言（换个角度，也许可以作为入门语言），因为很多曾经立志学习编程的人学了 C 语言之后，就决心不再学习编程了。难道是用 C 语言来筛选这个行业的从业者吗（从这个角度看就可以用 C 语言来入门了）？

但是，如果你要学习 Python，就不用担心所谓的基础问题。

特别是本书，就是强调“零基础”，这算是本书的特色之一。

不仅笔者这么认为，目前也有高校开始用 Python 作为软件专业甚至是非软件专业的大学生入门编程语言。某大学的教师（笔者的一名未曾谋面的网友）已经在教授经济学院的学生学习 Python 了。

最后的结论是：学习 Python，零基础足够。

本书的目标就是要跟你一起从零基础开始学习 Python，直到高手境界——不是笔者夸口，而是你要有信心。

所以，尽管放胆来学，不用犹豫、不要惧怕。还有一个原因，是因为她优雅。

0.2.3　从小工到专家

有不少学习 Python 的朋友询问：“书已经看了，书上的代码也运行过了，但是还不知如何开发一个真正的应用程序，不知从何处下手。”也遇到过一些大学毕业生，虽然相关专业的考试