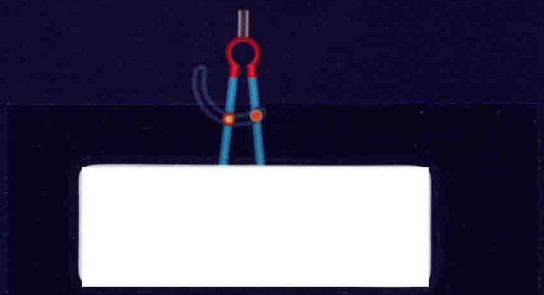
Broadview
www.broadview.com.cn

跟老齐学 Python 从入门到精通

齐伟 编著



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

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 从入门到精通

齐伟 编著

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

内 容 简 介

本书是面向编程零基础读者的 Python 入门教程，内容涵盖了 Python 的基础知识和初步应用。以比较轻快的风格，向零基础的学习者介绍一门时下比较流行、并且用途比较广泛的编程语言，所以，本书读起来不晦涩，并且在其中穿插了很多貌似与 Python 编程无关，但与学习者未来程序员职业生涯有关的内容。

本书特别强调了学习和使用 Python 的基本方法，学习一种高级语言，掌握其各种规则是必要的，但学会“自省”方法更重要，这也是本书所试图达到的“授人以鱼不如授人以渔”的目的。

本书是面向初学者的读物，不是为开发者提供的开发手册，所以，它不是“又适用于中高级读者”的读物。

图书在版编目（CIP）数据

跟老齐学 Python：从入门到精通 / 齐伟编著. —北京：电子工业出版社，2016.3
ISBN 978-7-121-28034-4

Ⅰ. ①跟… Ⅱ. ①齐… Ⅲ. ①软件工具－程序设计 Ⅳ. ①TP311.56

中国版本图书馆 CIP 数据核字（2016）第 007089 号

策划编辑：高洪霞
责任编辑：黄爱萍
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
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：25　字数：640 千字
版　　次：2016 年 3 月第 1 版
印　　次：2016 年 3 月第 1 次印刷
定　　价：69.00 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。

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。

服务热线：（010）88258888。

目　录

第 1 季　基础

第 2 季 进阶

第 3 季　实战

第1季

基　础

从这里开始，请读者——你已经确信自己是要学习 Python 的准程序员了——跟我一起，领略一番 Python 的基础知识，这是学好 Python 的起步，同时，其内容也和其他的高级编程语言有相通之处。所以，学习 Python 是一种“性价比”非常高的事情。

在本季中，要向读者介绍 Python 的基本对象类型、语法规则和函数的相关知识。学习完这些内容，就能够用 Python 做很多事情了，且在其中还会不断强化一种掌握 Python 的方法。

第0章

预备

从现在开始，本书将带领你——零基础的学习者——进入到 Python 世界。进入这个世界，你不仅能够体会到 Python 的魅力，感受到编程的快乐，还顺便可以成为一个程序员，我相信你一定能成为一个伟大的程序员，当然这并不是本书的目的，更不是本书的功劳。当你成为一个技术大牛的时候，最应该感谢的是你的父母，如果你顺便也感谢一下本书，比如多购买一些本书分发给你的弟兄们，那是我的福份，感激不尽。

预备，Let's go!

0.1 关于 Python 的故事

学习一种编程语言是一件很有意思的事情，从现在开始，我就和你一起来学习一种叫作 Python 的编程语言。

在编程界，存在着很多某种语言的忠实跟随者，因为忠实，就会如同卫道士一样有了维护那种语言荣誉的义务，所以总见到有人争论哪种语言好、哪种语言不好。当然，好与坏的标准是不一样的，有些人以学了之后能不能挣大钱为标准，有些人以是否容易学为标准，或许还有人以能不能将来和妹子一同工作为标准（也或许没有），甚至有些人就没有什么标准，只是凭感觉或者道听途说而人云亦云罢了。

读者在本书中将看到一个颇为迷恋于 Python 的人，因为全书看不到一句有关 Python 的坏话（如果有，则肯定是笔误，是应该删除的部分）。

不管是语言还是其他什么，挑缺点是比较容易的事情，但找优点都是困难的，所以，《圣经》中那句话——为什么你看见弟兄的眼中有刺，却不想自己眼中有梁木呢？——是值得我们牢记的。

在本书开始就废话连篇，显见本书不会有什么“干货”，倒是“水货”颇多，并不是因为“水是生命的源泉”，而是因为作者水平有限，如果不掺“水”，唯恐说不清道不明，还敬请读者谅解。嫌“水”多的，就此可以合上本书去看网上的各种电影吧。也不用在网上喷我，因为那样只能增加更多的“口水”（还是水）。

下面说点儿正经的。

0.1.1　Python 的昨天、今天和明天

这个题目有点大了，似乎回顾过去、考察现在、张望未来都是那些掌握方向的大人物做的。那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，似乎无法满足好奇心；如果不考察一下现在，也不放心（担心学了之后没有什么用途）；如果不张望一下未来，怎么能吸引（也算是一种忽悠吧）你呢？

1．Python 的历史

历史向来是成功者的传记，现在流传的关于 Python 的历史也是如此。

Python 的创始人为吉多·范罗苏姆（Guido van Rossum），关于他开发 Python 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，在吉多本人看来，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，也都转载了这一段文字。但是，在我来看，吉多是为了“打发时间”而决定开发 Python，源自他的这样一段自述：

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，这个哥们儿是一个非常牛的人，此处献上恭敬的崇拜。

其次，刚刚开始学习 Python 的朋友，可千万别认为 Python 是一个可以随随便便鼓捣的东西，人家也是站在巨人的肩膀上的。

第三，牛人在成功之后，往往把奋斗的过程描绘得比较简单。或者是出于谦虚，或者是为了让人听起来他更牛。反正，我们看最后结果的时候，很难感受过程中的酸甜苦辣。

不管怎样，吉多·范罗苏姆在那时刻创立了 Python，而且，更牛的在于他具有现代化的思维——开放，并通过 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呼和尊敬。

请读者向所有倡导“开放”的牛人们表示敬意，是他们让这个世界变得更美好，他们以行

动诠释了热力学第二定律——“熵增原理”。

2. Python 的现在

Python 现在越来越火了，因为它搭上了“大数据”、“云计算”、“自然语言处理”等这些时髦名词的便车。

网上时常会有一些编程语言排行榜之类的东西，有的初学者常常被排行榜所迷惑，总想要学习排列在第一位的，认为排在第一位的编程语言需求量大。不管排行榜是怎么编制的，Python 虽然没有登上状元、榜眼、探花之位，但也不太靠后呀。

另外一个信息，更能激动一下初学者那颗脆弱的小心脏。

Dice.com 网上对 20000 名 IT 专业人士进行调查的结果显示：Java 类程序员平均工资 91060 美元；Python 类程序员平均工资 90208 美元。

Python 程序员比 Java 程序员的平均工资低，但看看差距，再看看两者的学习难度，学习 Python 绝对是一个性价比非常高的投资。

这么合算的编程语言不学等待何时？

图 0-1　Python 创始人：吉多 • 范罗苏姆

3. Python 的未来

Python 的未来要靠读者了，你学好了、用好了，未来它就光明了，它的未来在你手里。如图 0-1 所示为 Python 创始人吉多 •范罗苏姆。

0.1.2　Python 的特点

很多高级语言都宣称自己是简单的、入门容易的，并且具有普适性，但真正能做到这些的，只有 Python。有朋友做了一件衬衫，上面写着“生命有限，我用 Python”，这说明什么？说明 Python 有着简单、开发速度快、节省时间和精力的特点。因为它是开放的，有很多可爱的开发者（为开放社区做贡献的开发者是最可爱的人），将常用的功能做好了放在网上，谁都可以拿过来使用。这就是 Python，这就是开放。

恭敬地抄录来自《维基百科》的描述：

Python 是完全面向对象的语言，函数、模块、数字、字符串都是对象，并且完全支持继承、重载、派生、多继承，有益于增强源代码的复用性。Python 支持重载运算符，因此也支持泛型设计。相对于 Lisp 这种传统的函数式编程语言，Python 对函数式设计只提供了有限的支持。有两个标准库（functools 和 itertools）提供了 Haskell 和 Standard ML 中久经考验的函数式程序设计工具。

虽然 Python 可能被粗略地分类为“脚本语言”（Script Language），但实际上一些大规模软件开发项目（例如 Zope、Mnet、BitTorrent 及 Google）也都广泛地使用它。Python 的支持者较喜欢称它为一种高级动态编程语言，原因是“脚本语言”泛指仅做简单程序设计任务的语言，如 shell script、VBScript 等，但其只能处理简单任务的编程语言，并不能与 Python 相提并论。

Python 本身被设计为可扩充的，并非所有的特性和功能都集成到语言核心。Python 提供了丰富的 API 和工具，以便程序员能够轻松地使用 C、C++、Cython 来编写扩充模块。Python 编译器本身也可以被集成到其他需要脚本语言的程序内。因此，很多人还把 Python 作为一种“胶水语言”（glue language）使用，使用 Python 将其他语言编写的程序进行集成和封装。在 Google 内部的很多项目，例如 Google Engine 使用 C++编写性能要求极高的部分，然后用 Python 或 Java/Go 调用相应的模块。《Python 技术手册》的作者马特利（Alex Martelli）说：“2004 年，Python 已在 Google 内部使用，Google 招募许多 Python 高手，但在这之前就已决定使用 Python。他们的目的是尽量使用 Python，在不得已时改用 C++；在操控硬件的场合使用 C++，在快速开发时使用 Python。”

可能这里面有一些术语还不是很理解，没关系，只要明白：Python 是一种很牛的语言，应用简单，功能强大，Google 都在使用，这就足够了，足够让你下决心学习了。

0.1.3　Python 哲学

Python 之所以与众不同，还在于它强调一种哲学理念：优雅、明确、简单。有一段诗歌读起来似乎很玄，但真实反映了 Python 开发者的开发理念：

```
The Zen of Python

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!
```

网上能够看到这段文字的中文译本，读者可以去搜索阅读。

0.2　从小工到专家

这个标题，我借用了一本书的名字——《程序员修炼之道：从小工到专家》，并在此特别推荐阅读。

“从小工到专家”也是很多刚学习编程的朋友的愿望。如何能实现呢？《程序员修炼之道：从小工到专家》这本书中，给出了非常好的建议，值得借鉴。

有一个学习 Python 的朋友曾问我：“书已经看了，书上的代码也运行过了，习题也能解答了，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，怎么办？”

另外，也遇到过一些刚刚毕业的大学生，从简历上看，相关专业的考试分数是不错的（我一般相信那些成绩是真的），但是，一讨论到专业问题，常常不知所云，特别是当让他面对真实的工作对象时，表现出来的比成绩单差太多了。

对于上述情况，我一般会武断地下一个结论：练得少。

要从小工成长为专家，必经之路是要多阅读代码，多调试程序。古言“拳不离手，曲不离口”，多练习是成为专家的唯一途径。

0.2.1 零基础

有一些初学者，特别是非计算机专业的人，担心自己基础差，不能学好 Python。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。但如果是“绝对零基础”也不用担心，本书就是从这个角度切入来满足你的需要的。凡事总得有一个开始，那么就让本书成为你学习编程语言的开始吧。

就我个人来看，Python 是比较适合作为学习编程的入门语言的。

美国有不少高校也这么认为，他们纷纷用 Python 作为编程专业甚至是非编程专业的大学生入门语言，如图 0-2 所示为美国各高校设立的编程语言专业。

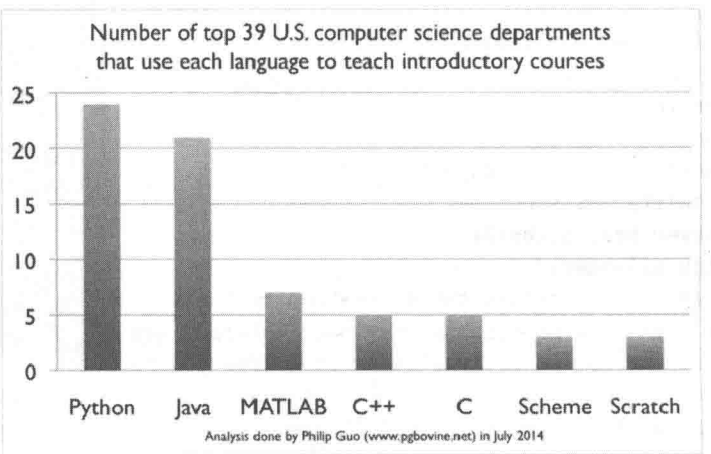


图 0-2 美国高校设立的编程语言专业

总而言之，学习 Python，你不用担心基础问题。

0.2.2 阅读代码

有句话说得好：“读书破万卷，下笔如有神”，这也适用于编程。通过阅读别人的代码，“站在巨人的肩膀上”，让自己眼界开阔，思维充实。

阅读代码的最好地方就是：www.github.com。

如果你还没有账号，请尽快注册，它将是你成为一个优秀程序员的起点。当然，不要忘记