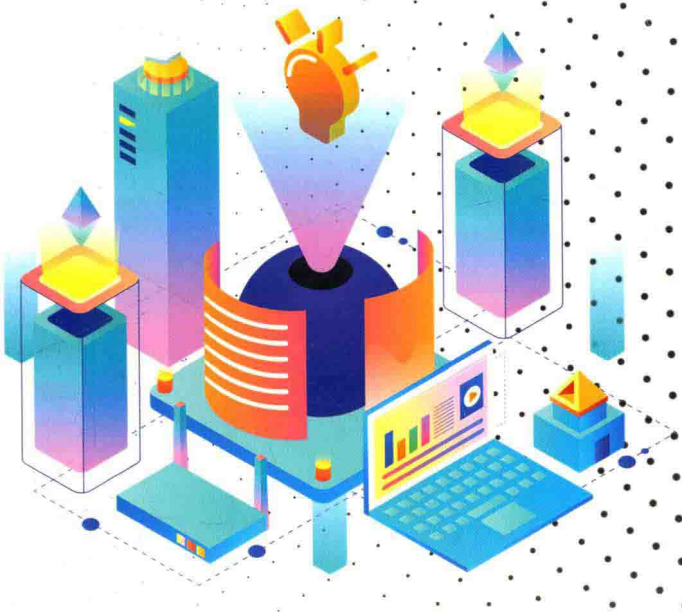
新工科建设之路·数据科学与大数据系列



Python

实用教程

刘宇宙 编著

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

新工科建设之路·数据科学与大数据系列

Python 实用教程

刘宇宙　编著

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

内 容 简 介

本书专门针对 Python 新手量身定做，是编者学习和使用 Python 开发过程中的体会和经验总结，涵盖实际开发中所有的重要知识点，内容详尽，代码可读性及可操作性强。本书主要介绍 Python 语言的类型和对象、操作符和表达式、编程结构和控制流、函数、序列、正则表达式、面向对象编程、文件操作等，各章还安排了活学活用、技巧点拨、问题探讨、章节回顾、实战演练等实例内容，以帮助读者学会处理程序异常、解答学习困惑、巩固知识、学以致用。本书使用通俗易懂的描述和丰富的实例代码，让复杂的问题以简单的形式展现出来，生动有趣，使读者学起来轻松，充分感受到学习 Python 的乐趣和魅力。

本书适合 Python 3.x 初学者，想学习和了解 Python 3.x 的程序员，Python 3.x 网课、培训机构、中学、大学本科、大专院校的学生，也可作为本、专科院校的教学用书。

图书在版编目（CIP）数据

Python 实用教程 / 刘宇宙编著. —北京：电子工业出版社，2019.5
ISBN 978-7-121-35884-5

Ⅰ. ①P… Ⅱ. ①刘… Ⅲ. ①软件工具－程序设计－高等学校－教材 Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字（2019）第 006525 号

策划编辑：章海涛
责任编辑：章海涛
印　　刷：三河市君旺印务有限公司
装　　订：三河市君旺印务有限公司
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：17.5　　字数：448 千字
版　　次：2019 年 5 月第 1 版
印　　次：2019 年 5 月第 1 次印刷
定　　价：42.00 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：192910558（QQ 群）。

前　言

以前有朋友问我 Python 的好处时，我用偏向于计算的语言跟他描述了一大堆，他好像仍然似懂非懂，最后问到，是不是 Python 的学习就像讲话一样。那一刻我突然意识到，对于没有接触过 Python 的学习者，我总想让他很快就融入到 Python 当中，殊不知对于他们来讲最容易理解的说法到底是怎样的。就像我的那位朋友，从他的角度来看，在我的描述下，它确实像讲话那样。

确实，Python 是一门编程语言，但它同时也像我们所说的话，非常灵活，每个人对它都可以有自己的学习方式，有自己的理解方式，有自己的操作方式。

我从来没有想过一门编程语言可以如此简单，只要你在计算机上花上几分钟构建好 Python 环境，就可以开始 Python 编程了。作为一门编程语言，它太适合零基础的朋友作为踏入编程大门的入门教程了。

Python 虽然简单，但语法结构非常严谨，就像我们说话，虽然可以用各种语言形式说，但也要保证说出内容的合理性，否则就很容易被误解或引起别人的愤怒。

Python 是一种解释型的、面向对象的、带有动态语义的高级程序设计语言。Python 中有很多术语，你可以在阅读本书的过程中逐渐弄懂。

Python 是一种使你在编程时能够保持自己风格的程序设计语言，使用它，你可以使用清晰易懂的程序来实现想要的功能。如果你之前没有任何编程经历，那么既简单又强大的 Python 就是你入门的完美选择。

Python 如何体现它的简单？有人说，完成一件相同的任务，使用汇编语言实现，可能需要编写 1000 行以上的代码，使用 C 语言实现，可能需要 500 行以上的代码，使用 Java 语言实现，可能需要 100 行以上的代码，而使用 Python 语言实现，可能只需要 20 行代码。这就是 Python，它可以帮你节约大量编写代码的时间。

从 Python 的市场需求看，随着人工智能、区块链、大数据、云计算、物联网等新兴技术的迅速崛起，市场对 Python 人才的需求和市场人才的匮乏让长期沉默的 Python 语言一下子备受众人的关注，本书可以说应运而生。

目前，Python 广泛使用的是 2.7 版本，新版本 Python 3 带来了很多新特性。本书是基于 Python 3.6 以上版本而编写的，在写作过程中，对所涉及的知识点，基本都使用的是当前最新的知识点，所以对于想学习 Python 的读者，完全不用担心在学习本书时，已经有很多新的知识点被更新了，从而担心自己又得去学习大量的新知识点。

本书特色

本书专门针对 Python 新手量身定做，是编者学习和使用 Python 开发过程中的体会和经验总结，涵盖实际开发中所有的重要知识点，内容详尽，代码可读性及可操作性强。

本书主要介绍 Python 语言的类型和对象、操作符和表达式、编程结构和控制流、函数、序列、正则表达式、面向对象编程、文件操作等，各章还安排了活学活用、技巧点拨、问题探讨、章节回顾、实战演练等内容，以帮助读者学会处理程序异常、解答学习困惑、巩固知识、学以致用。

本书的另一个特色是，使用通俗易懂的描述和丰富的示例代码，并结合日常生活中的一些小事件，使本书读起来尽可能生动有趣，让复杂的问题以简单的形式展现出来，使读者学起来轻松，充分感受到学习 Python 的乐趣和魅力。

本书通过 Python 快乐学习班的成员去往 Python 库的旅游贯穿全文，通过与现实中的旅游来和各个章节的知识点结合，让读者更加直观明了地理解各个章节的内容和知识点。

知识点与景点或服务区的对应如下：

进入 Python 世界——“数据类型”服务区

列表和元组——“序列号”接驳车

字符串——“字符串”主题游乐园

字典和集合——“字典屋”

条件、循环和其他语句——“循环旋转”乐园

函数——“函数乐高积木”

类与对象——“对象动物园”

异常处理——“异常过山车”

日期和时间——“时间森林”

正则表达式——“正则表达式寻宝古街”

文件——“文件魔法馆”

除了以旅游的形式展现知识内容，本书还基于 Python 的最新版本编写，书中所有示例都在最新的 Python 版本上运行成功，随书源码将发布在 Github 上。

本书内容

本书共分为 12 章，各章内容安排如下：

第 1 章主要介绍 Python 的起源、发展前景、Python 3 的一些新特性、环境构建及第一个 Python 程序。

第 2 章主要介绍 Python 的基础知识，讲解 Python 中的数据类型、变量和关键字、运算符和操作对象等的概念，为后续章节的学习做铺垫。

第 3 章主要介绍列表和元组，包括列表和元组的操作及两者的区别。

第 4 章主要介绍字符串，包括字符串的简单操作，格式化，字符串的方法等内容。

第 5 章主要介绍字典和集合，包括字典的创建和使用，字典方法，集合的使用等内容。

第 6 章主要介绍条件语句、循环语句及列表推导式等一些更深层次的知识点，包括 import 的使用、赋值操作、条件语句和循环等内容。

第 7 章主要介绍函数，函数是组织好的、可重复使用的、用来实现单一或相关联功能的代码段，本章围绕函数的定义、调用，函数的参数，变量的作用域等知识点展开讲解。

第 8 章主要介绍 Python 面向对象编程的特性，Python 从设计之初就是一门面向对象语言，它提供一些语言特性支持面向对象编程。本章围绕类的定义与使用、继承、多态、封装等知识点展开讲解。

第 9 章主要介绍异常，围绕异常定义、异常捕捉、异常处理、自定义异常等知识点展开讲解。

第 10 章主要介绍日期和时间，以 time 模块、datetime 模块、日历模块作为主要知识点展开讲解。

第 11 章主要介绍正则表达式，将通过 re 模块的方法的介绍逐步展开对 Python 中正则表达式使用的讲解。

第 12 章主要介绍文件，围绕操作文件、文件方法、序列化与反序列化等知识点展开讲解。

读者对象

Python 3.x 初学者。

想学习和了解 Python 3.x 的程序员。

Python 3.x 网课、培训机构、中学及高等学校本科、高职高专的学生。

关于本书

在本书写作之际，由我编写，清华大学出版社出版的《Python 3.5 从零开始学》《Python 3.7 从零开始学》这两本书在市场上已经获得很多读者的欢迎，但当我回头过来细看，这两本书中仍然有很多不够完善的地方，于是我便又重新编写了本书。

该书在编写过程中基本保持了前两本书的目录结构方式，但在内容上，做了非常大的改动。本书对之前讲解不够到位的地方做了更详尽的讲解，对之前一些章节安排上不合理的地方做了调整，对一些描述比较难以理解的地方做了更通俗的讲解。

本书包含配套教学课件、实例源代码，读者可登录华信教育资源网（www.hxedu.com.cn）免费下载。

致谢

虽然有前两本书的编写经验，但在本书写作过程中依然遇到了很多困难以及写作方式上的困惑，好在这是一个信息互联的时代，这让笔者有机会参阅很多相关文献资料，也让很多困难得以较好的解决。

在写作过程中参考了一些相关资源上的写作手法，这些资源上有一些技术点使用了非常形象生动的方式来阐述，参考的内容主要包括《Python 3.5 从零开始学》《Python 3.7 从零开始学》《Python 基础教程（第 2 版）》《笨办法学 Python（第 4 版）》《像计算机科学家一样思考 Python》、廖雪峰的博客以及 W3C 等资源。在此，对它们的编者表示真诚的感谢。

最后感谢《Python 3.5 从零开始学》《Python 3.7 从零开始学》读者们的鼓励和支持，正因为有你们通过 QQ、邮件、博客留言等方式不断指出书中的不足，不断提出问题与提出意见，才使得本书可以以一种更为通俗易懂的方式呈现出来。

CSDN 技术博客：youzhouliu

技术问答 E-mail：jxgzyuzhouliu@163.com

技术问答 QQ 群：700103920

随书源码地址：https://github.com/liuyuzhou/python/pythonsourcecode.git

目　录

第一章　Python 的自我介绍

本章主要介绍 Python 的起源、应用场合、发展前景，Python 3.x 与 Python 2.x 的区别，以及 Python 最新版本的一些新特性。本章还将介绍 Python 的环境构建。作为开门篇，将介绍一个简单的编写 Hello World 小程序。

1.1　Python 的起源

任何一门语言的出现都有它对应的创始人，Python 也不例外。

Python 的创始人为 Guido van Rossum（后文简称 Guido）。1982 年，Guido 从阿姆斯特丹大学获得数学和计算机硕士学位。尽管 Guido 算得上是一位数学家，不过他更享受计算机带来的乐趣。用 Guido 的话说，尽管他拥有数学和计算机双料资质，不过他趋向于做计算机相关的工作，并热衷于做所有和编程相关的活儿。

Guido 接触并使用过 Pascal、C、Fortran 等语言。这些语言的基本设计原则是让机器运行得更快。在 20 世纪 80 年代，虽然 IBM 和苹果已经掀起了个人计算机浪潮，但是那时候个人计算机的配置很低，比如早期的 Macintosh 只有 8MHz 的 CPU 主频和 128KB 的 RAM，一个大的数组就能占满内存，因此所有编译器的核心都是做优化，以便让程序能够运行。为了提高效率，程序员不得不像计算机一样思考，以便写出更符合机器口味的程序，在那个时代，程序员恨不得榨取计算机每一寸的能力，有人甚至认为 C 语言的指针是在浪费内存。至于动态类型、内存自动管理、面向对象等就不要想了，这些只会让你的计算机陷入瘫痪。

这种编程方式让 Guido 感到苦恼。虽然 Guido 知道如何用 C 语言写出一个功能，但整个编写过程却需要耗费大量时间。Guido 还可以选择 Shell，Bourne Shell 作为 UNIX 系统的解释器已经存在很久了。UNIX 的管理员常常用 Shell 写一些简单的脚本，以进行系统维护的工作，比如定期备份、文件系统管理等。在 C 语言中，许多上百行的程序在 Shell 中只用几行就可以完成。然而，Shell 的本质是调用命令，它不是一个真正的语言，比如 Shell 没有数值型的数据类型，运用加法运算都很复杂。总之，Shell 不能全面调动计算机的功能。

Guido 希望有一种语言能够像 C 语言一样全面调用计算机的功能接口，又可以像 Shell 一样轻松编程。ABC 语言让 Guido 看到了希望，该语言是由荷兰的数学和计算机研究所开发的，Guido 曾经在该研究所工作，并参与了 ABC 语言的开发。与当时大部分语言不同，ABC 语言是以教学为目的，目标是“让用户感觉更好”，希望通过 ABC 语言让语言变得容易阅读、容易使用、容易记忆、容易学习，并以此激发人们学习编程的兴趣。

ABC 语言尽管已经具备了良好的可读性和易用性，不过始终没有流行起来。当时，ABC

语言编译器需要配置比较高的计算机才能运行，而这些计算机的使用者通常精通计算机，他们考虑更多的是程序的效率，而不是学习难度。ABC 语言不能直接操作文件系统，尽管用户可以通过文本流等方式导入数据，不过 ABC 语言无法直接读写文件，输入输出的困难对于计算机语言来说是致命的。你能想象一款打不开车门的跑车吗？

1989 年，为了打发圣诞节假期，Guido 开始写 Python 语言的编译器。Python 这个名字来自于 Guido 所挚爱的电视剧——Monty Python's Flying Circus。他希望这个新语言 Python 能够符合他的理想：创造一种介于 C 和 Shell 之间，功能全面、易学易用、可拓展的语言。Guido 作为一个语言设计爱好者，已经尝试过设计语言，这次不过是一种纯粹的 hacking 行为。

1991 年，第一个 Python 编译器诞生。该编译器是用 C 语言实现的，并且能够调用 C 语言的库文件。Python 诞生时便具有类、函数、异常处理，包含表和词典在内的核心数据类型，以及以模块为基础的拓展系统。

Python 的很多语法来自于 C，却又受 ABC 语言的强烈影响。来自 ABC 语言的一些规定至今还富有争议（比如强制缩进），不过这些语法规定让 Python 容易理解。另一方面，Guido 聪明地选择让 Python 服从一些惯例，特别是 C 语言的惯例，比如回归等号赋值。Guido 认为“常识”确定的东西没有必要过度纠结。

Python 从一开始就特别在意可拓展性。Python 可以在多个层次上拓展，在高层可以直接引入.py 文件，在底层可以引用 C 语言的库。程序员可以使用 Python 快速编写的.py 文件作为拓展模块。当性能是重点考虑的因素时，程序员可以深入底层写 C 程序，将编译的.so 文件引入 Python 中使用。Python 就像使用钢筋建房一样，要先规定好大的框架，程序员可以在此框架下相当自由地拓展或更改。

最初，Python 完全由 Guido 本人开发，后来逐渐受到 Guido 同事的欢迎，他们迅速反馈使用意见，并参与 Python 的改进。Guido 和一些同事构成了 Python 的核心团队，他们将自己大部分业余时间用于 hack Python，Python 逐渐拓展到了研究所外。Python 将许多机器层面的细节隐藏交给编译器处理，并凸显逻辑层面的编程思考，程序员使用 Python 时，可以将更多时间用于程序逻辑的思考，而不是具体细节的实现，这一特征吸引了广大程序员，Python 开始流行起来了。

1.2 Python 的发展前景与应用场合

现在，全世界有 600 多种编程语言，但流行的编程语言也就 20 多种。如果你听说过 TIOBE 排行榜，就能知道编程语言的大致流行程度。图 1-1 是 2002－2018 年最常用的 10 种编程语言的变化图。

2015 年到 2017 年，Python 基本处于第 5 位，市场占有率次于 Java、C、C++和 C#，从 2017 年开始，Python 就借着人工智能的东风，热度一路水涨船高，目前已经到第 4 位，有些排名机构甚至将其排为第一。Python 是一门比较注重效率的语言，不复杂，读和写都非常方便，所以才有“人生苦短，我用 Python”这样的调侃。人工智能、云计算和大数据方向对 Python 人才的需求也在不断加大。

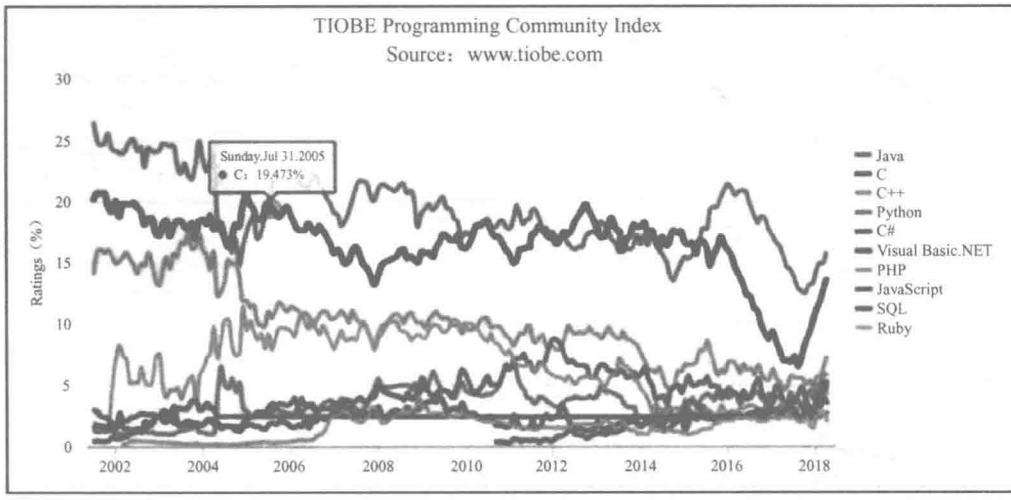


图 1-1 TIOBE 排行榜

当前 Python 使用最为广泛的领域是人工智能。在人工智能这块土地上，存在数不尽的未开垦的沃土，等待着每一位拓荒者的到来，当然，你需要先准备好足够好的“拓荒”工具——Python。

除人工智能领域外，区块链领域也有大量使用 Python 做具体实现的应用案例。

Python 在云计算方面的用途也很大，比如云计算中 IaaS（Infrastructure as a Service，基础设施即服务）层的很多软件都大量使用 Python，云计算的其他服务都建立在 IaaS 服务的基础上。

下面这些比较火热的软件中都大量使用 Python。

（1）Google 深度学习框架 TensorFlow 全由 Python 语言实现。

（2）Google 深度学习框架 Kares 全由 Python 语言实现。

（3）深度学习框架 Caffe 由 Python 语言实现。

（4）开源云计算技术（OpenStack）的源码全由 Python 语言实现。

（5）Amazon s3 命令行管理工具（s3cmd）。

（6）EC2 云计算管理工具（StarCluster）。

在大数据领域，Python 的使用也越来越广泛，Python 在数据处理方面有如下优势：

（1）异常快捷的开发速度，代码非常少。

（2）丰富的数据处理包，无论是正则，还是 HTML 解析、XML 解析，用起来都非常方便。

（3）内部类型使用成本很低，不需要许多额外操作（Java、C++用一个 Map 都很费劲）。

（4）公司中大量数据处理工作不需要面对非常大的数据。

（5）巨大的数据不是语言所能解决的，需要处理数据的框架（如 Hadoop）。Python 有处理大数据的框架，一些框架也支持 Python。

（6）编码问题处理起来非常方便。

除了在人工智能、区块链、云计算和大数据领域的应用，很多网站也是用 Python 开发的，很多大公司如 Google、Yahoo 和 NASA 都大量使用 Python。

我们熟知的 AlphaGo 就是 Google 用 TensorFlow 实现的，Facebook 也是扎克伯格用 Python 开发出来的，后来的 Twitter 也是用 Python 写的，实际上，Python 是国外很多大公司（如 Google）使用的主要语言。

Python 的定位如图 1-2 所示，为“优雅”“明确”“简单”。Python 程序看上去总是简单易懂，初学者学 Python 不但容易入门，而且将来深入下去可以编写非常复杂的程序。

图 1-2　Python 的定位

Python 的哲学就是简单、优雅、明确，尽量写容易看明白的代码，尽量将代码写得更少。

Python 是一个简单、解释型、交互式、可移植、面向对象的超高级语言。这是对 Python 语言最简单的描述。

Python 有一个交互式的开发环境，Python 的解释运行大大节省了每次编译的时间。Python 语法简单，内置几种高级数据结构（如字典、列表等），使用起来也特别简单。Python 具有大部分面向对象语言的特征，可完全进行面向对象编程。Python 可以在 MS-DOS、Windows、Windows NT、Linux、Solaris、Amiga、BeOS、OS/2、VMS、QNX 等多种操作系统上运行。

1.3　Python 的版本迭代

在编程语言的不断发展中，会不断有新的功能添加进来，也有一些不合理的功能做更改或被废除。对于一门编程语言，一直面临着不断迭代更新的问题。

目前，Python 有两个使用比较广泛的版本，一个是 2.x 版，一个是 3.x 版，这两个版本是不兼容的。3.x 不考虑对 2.x 代码的向后兼容。

当前有很多公司的项目还是基于 2.7 版本进行开发的，但是 3.x 版会越来越普及，特别是 Python 官方在 2018 年 6 月份宣布从 2020 年 1 月 1 日之后不再维护 Python 2.x 版本，这意味着 Python 2.x 将成为历史，所以本书将基于最新的 Python 3.x 版本进行后续内容的介绍。

在本书写作之时，Python 的最新版本是 3.7.0，本书中的所有示例和讲解内容都将基于这个版本进行。读者若想跟随书上示例一起练习，建议至少安装 3.6.1 以上的版本，对于 3.6 以下的版本，会出现一些示例的执行结果与书上描述结果不一致的情况，所以建议安装 3.7 版本，那样学习本书中的内容才会更加容易。

在 3.x 中，一些语法、内建函数和对象的行为有所调整。虽然大部分 Python 库都同时支持 Python 2.x 和 3.x 版本，无论选择哪个版本都可以成功执行，但为了在使用 Python 时避免某些版本中常见的陷阱，或者避免移植某个 Python 项目时遇到难题，依然有必要了解一下 Python 2.x 和 3.x 这两个常见版本之间的明显区别。

2.x 和 3.x 版本之间的明显区别如下：

1. 使用__future__模块

Python 3.x 引入了一些与 Python 2.x 不兼容的关键字和特性。在 Python 2.x 中，可以通过内置的__future__模块导入这些新内容。如果希望在 Python 2.x 中编写的代码也可以成功地在 Python 3.x 中执行，那么建议使用__future__模块。

2. print()函数

虽然 print 语法是 Python 3.x 中一个很小的改动，而且应该已经广为人知，但是依然值得一提：Python 2.x 中的 print 语句被 Python 3.x 中的 print()函数取代，由语句到函数的变化，

意味着在 Python 3.x 中必须用括号将需要输出的对象括起来。在 Python 2.x 中使用额外的括号也可以，但是如果在 Python 3.x 中写形式如 Python 2.x 中的 print 语句，就会触发 SyntaxError（语法错误）。

3. 整数除法

人们常常会忽视 Python 3.x 在整数除法上的改动，在 Python 3.x 中，对于整数除法，即使写错了也不会触发 SyntaxError，因此在移植代码或在 Python 2.x 中执行 Python 3.x 的代码时需要特别注意这个改动。

4. Unicode

Python 2.x 有基于 ASCII 的 str()类型，可通过单独的 unicode()函数转成 unicode 类型，但没有 byte 类型。在 Python 3.x 中，有了 Unicode（UTF-8）字符串和两个字节类（bytes 和 bytearrays）。

5. xrange

在 Python 2.x 中，经常会用 xrange()创建一个可迭代对象，通常出现在“for 循环”或“列表/集合/字典推导式”中。在 Python 3.x 中，range()函数的实现方式与 xrange()函数相同，所以 Python 3.x 中已经将 xrange()函数废除，若强行使用 xrange()函数，程序执行时将会触发 NameError 错误。

6. 触发异常

Python 2.x 支持带括号和不带括号的两种异常触发的语法，而 Python 3.x 只支持带括号的语法，对于不带括号的语法，执行时会触发 SyntaxError 错误。

7. 处理异常

在 Python 3.x 中，处理异常必须使用 as 关键字，而在 Python 2.x 中不需要使用 as 关键字。

8. next()函数和.next()方法

在实际的应用中，next()函数（.next()方法）会被经常使用到，因此这里需要着重提一下 next()的语法改动（实现方面也做了改动）：在 Python 2.x 中，next()的函数形式和方法形式都可以使用；在 Python 3.x 中，只能使用 next()函数，若强行调用.next()方法将会触发 AttributeError 的错误（此处区别在 next 前面是否加一点，不加一点的是函数，加一点的是方法）。

9. 使用 input()解析输入内容

Python 3.x 改进了 input()函数，改进后的函数总是将用户的输入存储为 str 对象。在 Python 2.x 中，因为会发生读取非字符串类型的一些危险行为，不得不使用 raw_input()代替 input()。

10. 返回可迭代对象，而不是列表

某些函数和方法在 Python 3.x 中返回的是可迭代对象，而不像在 Python 2.x 中返回列表。对象只遍历一次会节省很多内存，如果通过生成器多次迭代这些对象，效率就不高了。此时如果需要列表对象，可以通过 Python 3.x 的 list()函数简单地将可迭代对象转成列表。

当前最新的 Python 3.7 版本有如下新特性：

（1）添加了对 async for 在 list、set、dict 解析式以及 generator 表达式中的使用支持。

（2）支持 nanosecond 的时间函数，方便对 nanosecond 的操作，提供了 6 个新增的函数，分别为：clock_gettime_ns()，clock_settime_ns()，monotonic_ns()，perf_counter_ns()，process_time_ns() 和 time_ns()。

（3）在新版本中添加了@dataclass 装饰器，利用该装饰器可以减少数据类型定义的代码行数。

1.4 如何学习 Python

随着前面几本书的出版，有越来越多的同学问我 Python 如何学习，他们中有计算机专业出身的，有非计算机专业出身的，有已经开始学习 Python，但仍然感觉迷茫的，有准备进入 Python 学习的。

千里之行，始于跬步。Python 是一门编程语言，同时也是一门实践性的学科。对于实践性的学科，练习是最重要的，并且最好能跟着书中示例，一步一个脚印地进行学习。不要急于看完本书，前面先慢慢掌握基本知识点，基础知识掌握稳固后，后面就可以逐步加速，甚至按照自己的兴趣点去查阅相关书籍或网站资料。

在学习的过程中，对于遇到的例子最好能逐步形成自己先思考的习惯，思考后再看看给出的示例是怎样的，在这个过程中或许能找到比示例更好的处理方法。

在写代码时，千万不要用“复制”“粘贴”，把代码从其他地方粘贴到你的代码编译器里。写程序讲究一个感觉，需要一个字母一个字母地把代码敲进去。在敲代码的过程中，初学者经常会敲错，敲错后需要通过仔细检查、对照方能把错误定位，错误查找的过程，其实是在找一种适合自己思维逻辑的过程，这种思维逻辑一旦形成，就为后面遇到的大部分问题找到了一种最便捷的路径，这样就能以最快的速度掌握如何写程序。在编写代码的过程中，宁可写得慢或多写几遍，刚开始学习或许很吃力，但随着慢慢积累和熟悉，后面会越来越快，越来越顺畅。

就是在开始的阶段，要学会面对失败、未知和不断碰壁，当遇到的失败次数多了，未知逐步变成熟悉，碰壁碰得感觉痛了，后面再遇到类似问题，就知道怎么处理、怎么解决、怎么避免。

Python 作为一门不断发展与普及的语言，还在不断更新中。如果要了解有关最新发布的版本和相关工具的内容，http://www.python.org 就是一个聚宝盆。加入一些 Python 学习社区或找到一些有共同爱好的人一起学习交流是非常好的学习 Python 的方式。正所谓集思广益，一起思考与学习的人多了，读者能接触和学到的知识就会更多。在互联网时代，更应该发挥网络的作用，通过网络学习更新颖、更与时俱进的知识。

语言的发展总是在不断变化，一门语言最好的学习方式就是持续不断地去使用，不断地去更新自己的知识库。语言本身会不断地更新，学习者也要不断学习新的知识点，时刻保持与时俱进、跟上语言的发展。

以下网址可以帮助读者更好地学习 Python：

（1）http://www.liaoxuefeng.com/。

（2）http://www.runoob.com/python3/python3-tutorial.html。

1.5 Python 安装

工欲善其事，必先利其器。在开始编程前，需要先准备好相关工具。下面简要介绍如何

下载和安装 Python。

Python 的安装软件可以从 Python 官方网站下载，地址：https://www.python.org/downloads/。建议下载软件时从对应的官方网站下载，这样比较权威，而且更加安全。

1.5.1 在 Windows 系统中安装 Python

在 Windows 系统中安装 Python 可以参照以下步骤。

（1）打开 Web 浏览器（如百度浏览器、Google、火狐等），访问 https://www.python.org/downloads/，进入网页，可以看到如图 1-3 所示的页面，单击图中白色按钮，进入对应的软件下载页面，即可进行软件下载。

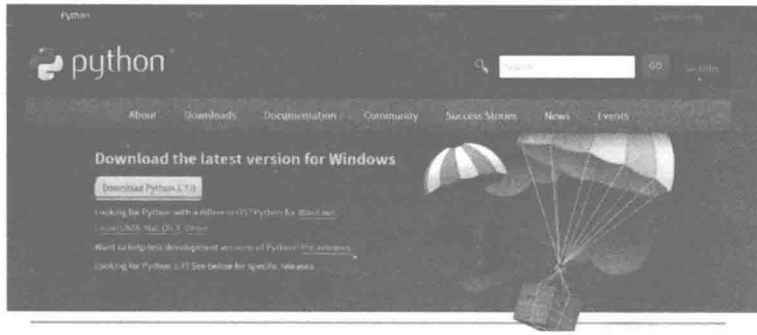


图 1-3　Python 官方网站下载页面

（2）下载安装软件后，接下来进行软件的安装。

① 双击下载好的软件，或者选中并右击下载好的软件，在弹出的对话框中选择“打开”选项，可以看到如图 1-4 所示的界面。底部的第一个复选框默认自动勾选，保持勾选状态即可，Add Python 3.7 to PATH 复选框默认不勾选，需要手动勾选，可以将 Python 的安装路径添加到环境变量中，勾选后，后面可省去该操作。如果希望将 Python 安装到指定路径下，就单击 Customize installation。如果单击 Install Now，系统就会直接开始安装 Python，并安装到默认路径下（此处建议安装到指定的目录）。

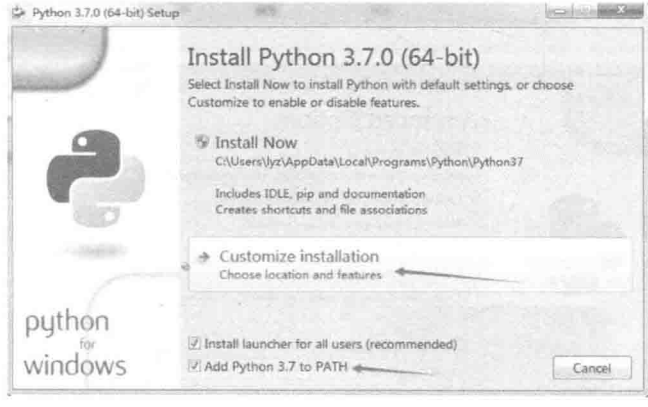


图 1-4　安装 Python

② 单击 Customize installation 后，会看到如图 1-5 所示的界面。此处直接单击 Next 按钮即可。