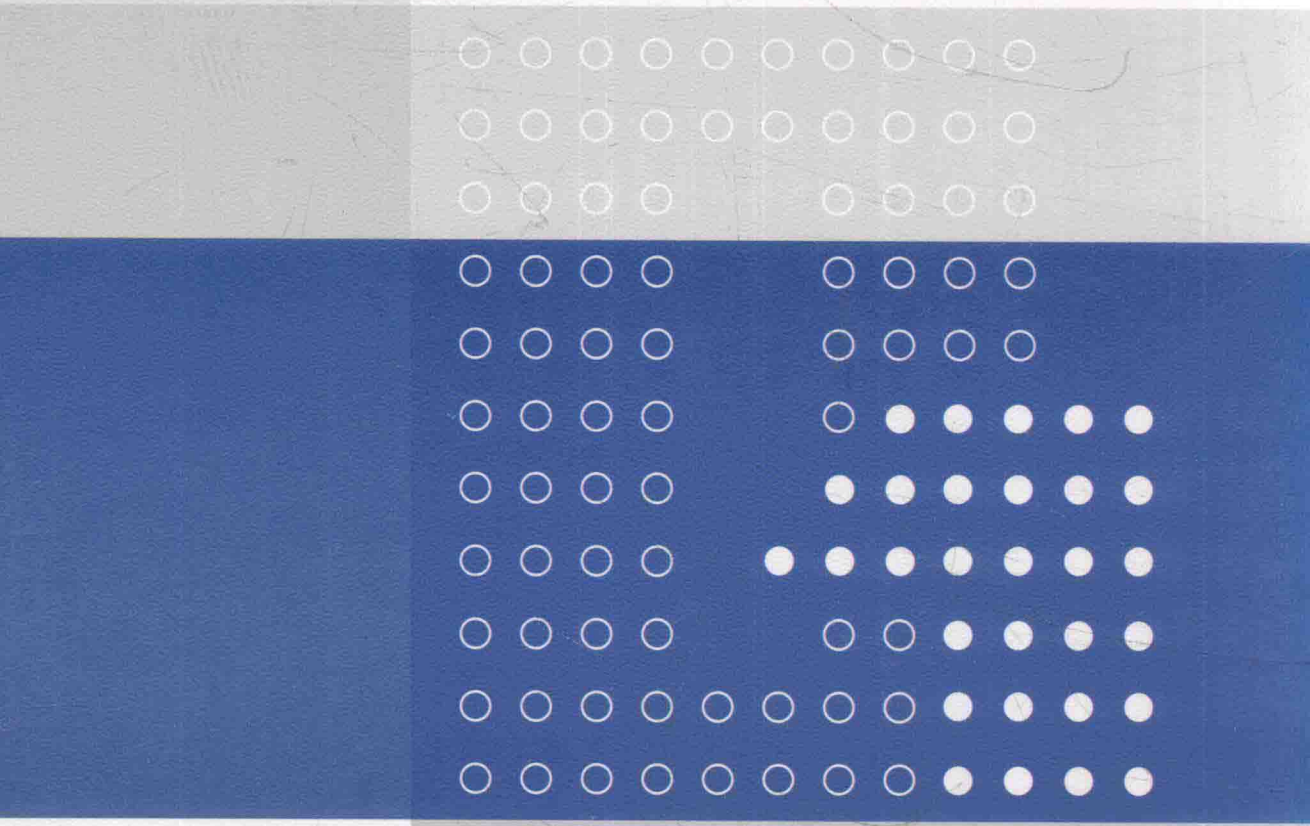
计算机系列教材

Python程序设计基础
(第2版)



董付国　编著

清华大学出版社

计算机系列教材

董付国　编著

Python程序设计基础
(第2版)

清华大学出版社
北京

内 容 简 介

全书共 13 章：第 1 章介绍 Python 基本知识与概念，Python 开发环境配置与使用，扩展库安装与使用；第 2 章讲解 Python 运算符与表达式以及内置函数的用法；第 3 章讲解列表、元组、字典、集合等序列结构的常用方法和基本操作；第 4 章讲解 Python 选择结构与循环结构的语法和应用；第 5 章讲解函数的定义与使用，不同类型的函数参数，变量的作用域以及 lambda 表达式；第 6 章讲解类的定义与实例化，多种不同类型的成员方法，特殊方法与运算符重载；第 7 章讲解字符串对象及其方法的应用；第 8 章讲解正则表达式语法以及正则表达式在 Python 中的应用；第 9 章讲解文件操作的基本知识与 Python 文件对象，文本文件内容读写，二进制文件操作与对象序列化；第 10 章讲解文件复制、移动、重命名、遍历等文件级操作以及目录操作有关知识；第 11 章讲解 Python 中多种不同形式的异常处理结构；第 12 章讲解 Python 对 SQLite 以及 Access、MS SQL Server、MySQL 等不同数据库的操作；第 13 章讲解数据分析、数据处理、数据可视化以及科学计算的有关知识。

本书完全面向 Python 3.x，全部案例代码使用 Python 3.5.x 和 Python 3.6.x 编写，大部分内容也同样适用于 Python 3.4.x。本书对 Python 内部工作原理进行一定程度的剖析，并适当介绍了 Python 代码优化和安全编程的有关知识，可以满足不同层次读者的需求。

图书在版编目(CIP)数据

Python 程序设计基础/董付国编著. —2 版. —北京：清华大学出版社，2018
(计算机系列教材)
ISBN 978-7-302-49056-2

Ⅰ. ①P…　Ⅱ. ①董…　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(2017)第 296436 号

责任编辑：白立军
封面设计：常雪影
责任校对：白　蕾
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954

印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：19.5　　**字　　数**：451 千字
版　　次：2015 年 8 月第 1 版　2018 年 1 月第 2 版　　**印　　次**：2018 年 1 月第1 次印刷
印　　数：1～2000
定　　价：49.00 元

产品编号：077590-01

前 言

FOREWORD

Python 由 Guido van Rossum 于 1989 年底开始研制，第一个版本发行于 1991 年。Python 推出不久就迅速得到了各行业人士的青睐，经过近 30 年的发展，已经渗透到计算机科学与技术、统计分析、逆向工程与软件分析、电子取证、图形图像处理、人工智能、游戏设计与策划、网站开发、移动终端开发、大数据分析与处理、深度学习、科学计算可视化、云计算、网络爬虫开发、系统运维、自然语言处理、密码学、电子电路设计、树莓派开发等专业和领域。目前，Python 已经成为卡耐基梅隆大学、麻省理工学院、加州大学伯克利分校、哈佛大学等国外很多大学计算机专业或非计算机专业的程序设计入门教学语言，国内也有不少学校的多个专业陆续开设了 Python 程序设计课程。

Python 连续多年在 TIOBE 网站的编程语言排行榜上排名前十位，并于 2011 年 1 月被 TIOBE 网站评为 2010 年度语言。自 2015 年之后，Python 一直稳居 TIOBE 编程语言排行榜前五位。在 2014 年 12 月份 IEEE Spectrum 推出的编程语言排行榜中，Python 排第 5 位，而在 2017 年 7 月份 IEEE Spectrum 推出的编程语言排行榜中，Python 上升到了第 1 位。

Python 是一门免费、开源的跨平台高级动态编程语言，支持命令式编程、函数式编程，完全支持面向对象程序设计，语法简洁清晰，并且拥有大量功能强大的标准库和扩展库以及众多狂热的支持者，可以帮助各领域的科研人员或策划师甚至管理人员快速实现和验证自己的思路与创意。Python 用户可以把主要精力放在业务逻辑的设计与实现上，而不用过多考虑语言本身的细节，开发效率非常高，其精妙之处令人击节叹赏。

Python 是一门快乐的语言，学习和使用 Python 也是一个快乐的过程。与 C 语言系列和 Java 等语言相比，Python 更加容易学习和使用，但这并不意味着可以非常轻松愉快地掌握 Python。用户熟练掌握和运用 Python 仍需要通过大量的练习来锻炼自己的思维和熟悉 Python 编程模式，同时还需要经常关注 Python 社区优秀的代码以及各种扩展库的最新动态。当然，如果能够适当了解 Python 标准库以及扩展库的内部工作原理，对于编写正确而优雅的 Python 程序无疑是有很大帮助的。

Python 是一门优雅的语言。Python 语法简洁清晰，并且提供了大量的内置对象和内置函数，编程模式非常符合人类的思维方式和习惯。在有些编程语言中需要编写大量代码才能实现的功能，在 Python 中仅需要调用内置函数或内置对象的方法即可实现。如果读者已有其他程序设计语言的基础，那么在学习和使用 Python 时，一定不要把其他语言的编程习惯和风格带到 Python 中来，因为这不仅可能会使得代码变得非常冗余、烦琐，还可能会严重影响代码的运行效率。应该尽量尝试从最自然、最简洁的角度出发去思考和解决问题，这样才能写出更加优雅、更加纯正、更加 Pythonic 的代码。

本书内容组织

对于 Python 程序员来说,能够熟练运用各种扩展库毫无疑问是非常重要的,使用优秀、成熟的扩展库可以帮助我们快速实现自己的业务逻辑和创意。但是也必须清楚地认识到,Python 内功是非常重要的,Python 语言基础知识和基本数据结构的熟练掌握是理解和运用其他扩展库的必备条件之一。所以,本书前 11 章把重点和主要篇幅放在了 Python 编程基础知识的介绍上,通过大量案例介绍 Python 在实际开发中的应用,然后在最后两章介绍数据库编程和 Python 在数据分析、处理与科学计算可视化等领域的应用。关于其他应用领域的扩展库可以参考本书最后的附录,并结合自己的专业领域查阅相关文档。全书共 13 章,主要内容组织如下。

第 1 章　管中窥豹:Python 概述。介绍 Python 语言的特点,Python 程序文件名,扩展库的管理与使用,Python 代码编写规范和优化建议。

第 2 章　万丈高楼平地起:运算符、表达式与内置对象。讲解 Python 对象模型,数字、字符串、列表、元组、字典、集合等基本数据类型,运算符与表达式,内置函数。

第 3 章　玄之又玄,众妙之门:详解 Python 序列结构。讲解列表、元组、字典、集合等序列的常用方法和基本操作,切片操作,列表推导式,元组与生成器推导式,序列解包,字典、集合基本操作和常用方法。

第 4 章　反者,道之动:程序控制结构。讲解 Python 选择结构,for 循环与 while 循环,带有 else 子句的循环结构,break 与 continue 语句,选择结构与循环结构的综合运用。

第 5 章　代码复用技术(一):函数。讲解函数的定义与使用,普通位置参数、关键参数、默认值参数、长度可变参数等不同参数类型,全局变量与局部变量,参数传递时的序列解包,return 语句,lambda 表达式。

第 6 章　代码复用技术(二):面向对象程序设计。讲解类的定义与继承,self 与 cls 参数,类成员与实例成员,私有成员与公有成员,特殊方法与运算符重载。

第 7 章　文本处理(一):字符串。讲解字符串编码格式,字符串格式化、替换、分割、连接、排版等基本操作方法。

第 8 章　文本处理(二):正则表达式。讲解正则表达式语法、正则表达式对象、子模式与 match 对象,以及 Python 正则表达式模块 re 的应用。

第 9 章　数据永久化:文件内容操作。讲解文件操作基本知识与 Python 文件对象,文本文件内容读写,二进制文件内容读写与对象序列化,Word、Excel 等常见二进制文件的内容读写。

第 10 章　文件与文件夹操作。讲解文件复制、移动、重命名、遍历等文件级操作以及目录操作有关知识。

第 11 章　代码质量保障:异常处理结构与单元测试。讲解 Python 异常类层次结构与自定义异常类,多种不同形式的异常处理结构,以及单元测试。

第 12 章　数据库应用开发。讲解 SQLite 数据库的基本特点与用法,以及 Python 对

SQLite 数据库和 Access、MySQL、MS SQL Server 等数据库的操作方法。

第 13 章　数据分析与科学计算可视化。讲解 Python 标准库 statistics 以及 numpy、scipy、pandas、matplotlib 等扩展库的用法，讲解数据处理、数据分析、数据可视化以及科学计算的有关内容。

本书特色

内容与 Python 最新版本同步。本书完全面向 Python 3. x，全部案例代码使用 Python 3. 5. x 和 Python 3. 6. x 编写，大部分内容同样适用于 Python 3. 4. x。

信息量大、知识点密集。全书没有多余的文字和软件安装截图，充分利用宝贵的篇幅来介绍和讲解尽可能多的知识点，绝对物超所值。本书作者具有 15 年程序设计教学经验，讲授过汇编语言、C/C++ /C#、Java、PHP、Python 等多门程序设计语言，并编写过大量的应用程序。在本书内容的组织和安排上，结合了作者多年教学与开发过程中积累的许多案例，并巧妙地糅合进了相应的章节。

案例丰富，实用性强，注释量大。精选多个领域中的经典案例，并且每段代码都配有大量注释，大幅度缩短了读者理解代码所需要的时间。

语言精练，代码优雅。使用最简练的语言和代码介绍 Python 语法和应用，完美诠释 Pythonic 真谛。

深度与广度兼顾。本书对 Python 内部工作原理进行一定程序的剖析，并适当介绍 Python 代码优化和安全编程的有关知识，可以满足不同层次读者的需要，读者对书中内容每多读一遍都会有新的收获和体会。

本书适用读者

本书可以作为(但不限于)：

- 会计、经济、金融、心理学、统计、管理、人文社科以及其他非计算机专业本科或专科的程序设计教材。如果作为本科非计算机专业程序设计语言公共课或选修课教材，建议采用 64 学时或 48 学时边讲边练的教学模式。
- 具有一定 Python 基础的读者进阶学习资料。
- 打算利用业余时间学习一门快乐的程序设计语言并编写几个小程序来娱乐的读者首选学习资料。
- 少数对编程具有浓厚兴趣和天赋的中学生课外阅读资料。

教学资源

本书提供全套教学课件、源代码、课后习题答案与分析、考试题库、教学视频、教案以及授课计划和学时分配表，需要配套资源，可以登录清华大学出版社官方网站(www. tup. com. cn)下载或与作者联系索取，作者的微信公众号是“Python 小屋”，电子邮箱地址是 dongfuguo2005@126. com。

由于时间仓促，作者水平有限，书中难免存在疏漏之处，还请同行指正并通过作者联

系方式进行反馈,作者将不定期在微信公众号更新勘误并实名感谢。

感谢

首先感谢父母的养育之恩,在当年那么艰苦的条件下还坚决支持我读书,而没有让我像其他同龄的孩子一样辍学。感谢姐姐、姐夫多年来对我的爱护以及在老家对父母的照顾,感谢善良的弟弟、弟媳在老家对父母的照顾,正是有了你们,我才能在远离家乡的城市安心工作。感谢我的妻子在生活中对我的大力支持,也感谢懂事的女儿在我工作时能够在旁边安静地读书而尽量不打扰我,并在定稿前和妈妈一起帮我阅读全书并检查出了几个错别字。

感谢每一位读者,感谢您在茫茫书海中选择了这本书,衷心祝愿您能够从本书中受益,学到您需要的知识!同时也期待每一位读者的热心反馈,随时欢迎指出书中的不足!

本书在编写出版过程中得到清华大学出版社的大力支持和帮助,在此表示衷心的感谢。

董付国　于山东烟台

2017 年 10 月

目 录

CONTENTS

第1章 管中窥豹：Python概述

1.1 Python是这样一种语言

有不少人说Python是一种“大蟒蛇语言”。虽然在英语中Python确实有大蟒蛇的意思，但Python语言和大蟒蛇却没有任何关系。Python语言的名字来自一个著名的电视剧*Monty Python's Flying Circus*，Python之父Guido van Rossum是这部电视剧的狂热爱好者，所以把他设计的语言命名为Python。

也有人说Python是一门脚本语言，这也不准确，远远不足以反映Python的强大。Python并不仅仅是一门脚本语言，更是一门跨平台、开源、免费的解释型高级动态编程语言，是一种通用编程语言。除了可以解释执行之外，Python还支持将源代码伪编译为字节码来优化程序提高加载和运行速度并对源代码进行保密，也支持使用py2exe、pyinstaller、cx_Freeze或其他类似工具将Python程序及其所有依赖库打包成为各种平台上的可执行文件，当然也包括扩展名为exe的Windows可执行程序，从而可以脱离Python解释器环境和相关依赖库，能够在Windows平台上独立运行，并且还支持制作成.msi安装包；Python支持命令式编程(How to do)和函数式编程(What to do)两种方式，完全支持面向对象程序设计，语法简洁清晰，功能强大且易学易用，更重要的是拥有大量的几乎支持所有领域应用开发的成熟扩展库和狂热支持者。

当然，也有人喜欢把Python称为“胶水语言”，这确实是Python的重要特点之一。它可以把多种不同语言编写的程序融合到一起实现无缝拼接，更好地发挥不同语言和工具的优势，满足不同应用领域的需求。

1.2 Python版本之争

众所周知，Python官方网站同时发行和维护着Python 2.x和Python 3.x两个不同系列的版本，并且版本更新速度非常快(6个月左右更新一次小版本号)。目前最新版本分别是Python 2.7.14、Python 3.4.7、Python 3.5.4和Python 3.6.3，Python 3.7已在研发中，估计很快就会推出。Python 2.x和Python 3.x这两个系列的版本之间很多用法是不兼容的(让人欣慰的是，除了一些新特性、运算符和标准库对象之外，同一个系列的不同版本之间绝大多数用法是完全一致的)，除了基本输入输出方式有所不同，很多内置函数和标准库对象的用法也有非常大的区别。Python 3.x在增加了很多新标准库的同时也

删除了一些 Python 2. x 的标准库,还有些 Python 2. x 的标准库在 Python 3. x 中进行了合并和拆分。当然,适用于 Python 2. x 和 Python 3. x 的扩展库之间更是差别巨大,这应该是现有系统进行版本迁移时的最大障碍。所以,在正式开始使用 Python 之前,必须要选择合适的版本,以免浪费时间。

总体来看,Python 3. x 的设计理念更加合理、高效和人性化,全面普及和应用是必然的,越来越多的扩展库也以非常快的速度推出了与最新 Python 版本相适应的版本。如果暂时还没想到要做什么行业领域的应用开发,或者仅仅是为了尝试一种新的、好玩的语言,那么请毫不犹豫地选择 Python 3. x 系列的最高版本。

1.3 Python 编程规范与代码优化建议

没有规矩,不成方圆。任何一种语言都有一些约定俗成的编码规范,Python 也不例外。Python 非常重视代码的可读性,对代码布局和排版有更加严格的要求。虽然一些大型软件公司对自己公司程序员编写的代码在布局、结构、标识符命名等方面有一些特殊的要求,但其中很多思想是相同的,目的也是一致的。这里重点介绍 Python 社区对代码编写的一些共同的要求、规范和一些常用的代码优化建议,最好在开始编写第一段代码时就遵循这些规范和建议,养成一个好习惯。

(1) 严格使用缩进来体现代码的逻辑从属关系。Python 对代码缩进是硬性要求,这一点必须时刻注意。如果某个代码段的缩进不对,那么整个程序就是错的,要么是语法错误无法执行,要么是逻辑错误导致错误结果,而检查这样的错误会花费很多时间。

(2) 每个 import 语句只导入一个模块,最好按标准库、扩展库、自定义库的顺序依次导入。尽量避免导入整个库,最好只导入确实需要使用的对象,这会让程序运行更快。

(3) 最好在每个类、函数定义和一段完整的功能代码之后增加一个空行,在运算符两侧各增加一个空格,逗号后面增加一个空格。按照这样的规范写出来的代码布局和排版比较松散,阅读起来更加轻松。不论是前面第一条讲的缩进,还是这里谈的空行与空格,主要是提高代码可读性,正如 *The Zen of Python* 所说:Sparse is better than dense, readability counts。稍微有点例外的是,在正常的赋值表达式中等号两侧都是各增加一个空格,但在定义函数的默认值参数和使用关键参数调用函数时一般并不在参数赋值的等号两侧增加空格。这样松中有紧也是为了提高代码的可读性,正所谓:"张而不弛,文武弗能也;弛而不张,文武弗为也;一张一弛,文武之道也。"

(4) 尽量不要写过长的语句。如果语句过长,可以考虑拆分成多个短一些的语句,以保证代码具有较好的可读性。如果语句确实太长而超过屏幕宽度,最好使用续行符(line continuation character)"\",或者使用圆括号将多行代码括起来表示是一条语句。

(5) 虽然 Python 运算符有明确的优先级,但对于复杂的表达式建议在适当的位置使用括号使得各种运算的隶属关系和计算顺序更加明确,正如 *The Zen of Python* 所说:Explicit is better than implicit。

(6) 对关键代码和重要的业务逻辑代码进行必要的注释。统计数据表明,一个可读性较好的程序中应包含大概 30%以上的注释。在 Python 中有两种常用的注释形式:#

和三引号。#用于单行注释，三引号常用于大段说明性文本的注释。

(7) 在开发速度和运行速度之间尽量取得最佳平衡。内置对象运行速度最快，标准库对象次之，用 C 或 FORTRAN 编写的扩展库速度也比较快，而纯 Python 的扩展库往往速度慢一些。所以，在开发项目时，应优先使用 Python 内置对象，其次考虑使用 Python 标准库提供的对象，最后考虑使用第三方扩展库。然而，有时候只使用内置对象和标准库对象的话，很可能无法直接满足需要。这时候有两个选择：一是使用内置对象和标准库对象编写代码实现特定的逻辑；二是使用合适的扩展库对象。至于如何取舍，最终还是取决于业务逻辑的复杂程度和对运行速度的要求这两者之间的平衡。

(8) 根据运算特点选择最合适的数据类型来提高程序的运行效率。如果定义一些数据只是用来频繁遍历，最好优先考虑元组或集合。如果需要频繁地测试一个元素是否存在于一个序列中并且不关心其位置，尽量采用字典或者集合。列表和元组的 in 操作的时间复杂度是线性的，而对于集合和字典却是常数级的，与问题规模几乎无关。在所有内置数据类型中，列表的功能最强大，但开销也最大，运行速度最慢，应慎重使用。作为建议，应优先考虑使用集合和字典，元组次之，最后考虑列表和字符串。

(9) 充分利用关系运算符以及逻辑运算符 and 和 or 的惰性求值特点，合理组织条件表达式中多个条件的先后顺序，减少不必要的计算。

(10) 充分利用生成器对象或类似迭代对象的惰性计算特点，尽量避免将其转换为列表、元组等类型，这样可以减少对内存的占用，降低空间复杂度。

(11) 减少内循环中的无关计算，尽量往外层提取。

有很多成熟的工具可以检查 Python 代码的规范性，如 pep8、flake8、pylint 等。可以使用 pip 来安装 pep8 工具，然后使用命令 pep8 test.py 来检查 test.py 文件中 Python 代码的规范性。pep8 常用的可选参数有--show-source、--first、--show-pep8 等。flake8 结合了 pyflakes 和 pep8 的特点，可以检查更多的内容，优先推荐使用，使用 pip install flake8 可以直接安装，然后使用命令 flake8 test.py 检查 test.py 中代码的规范性。也可以使用 pip 安装 pylint，然后使用命令行工具 pylint 或者可视化工具 pylint-gui 来检查程序的规范性。

1.4　Anaconda3 开发环境的安装与使用

Python 的开发环境非常多，可以根据自己的使用习惯进行选择。除了 Python 官方网站提供的 IDLE 开发环境，还有 PyCharm、wingIDE、PythonWin、Eclipse+PyDev、Eric。另外，为了方便使用 Python，Anaconda、Python(x,y)、zwPython 等安装包集成了大量常用的 Python 扩展库，大幅度节约了用户配置 Python 开发环境的时间。本书选择了目前在教学和科研中使用较多的 Anaconda3，但这并不是必需的，书中代码同样适用于其他开发环境。

登录网址 https://www.continuum.io/downloads 下载 Anaconda3 并安装之后，“开始”菜单中会增加图 1-1 显示的菜单，其中 Jupyter Notebook 和 Spyder 是使用较多的两个开发环境。启动 Jupyter Notebook 之后，在右上角单击 New，然后选择 Python [default](见图 1-2)进入交互式开发环境，在单元格内输入代码块后单击图 1-3 中箭头所指的按钮即可运行输入的代码并立刻得到结果。