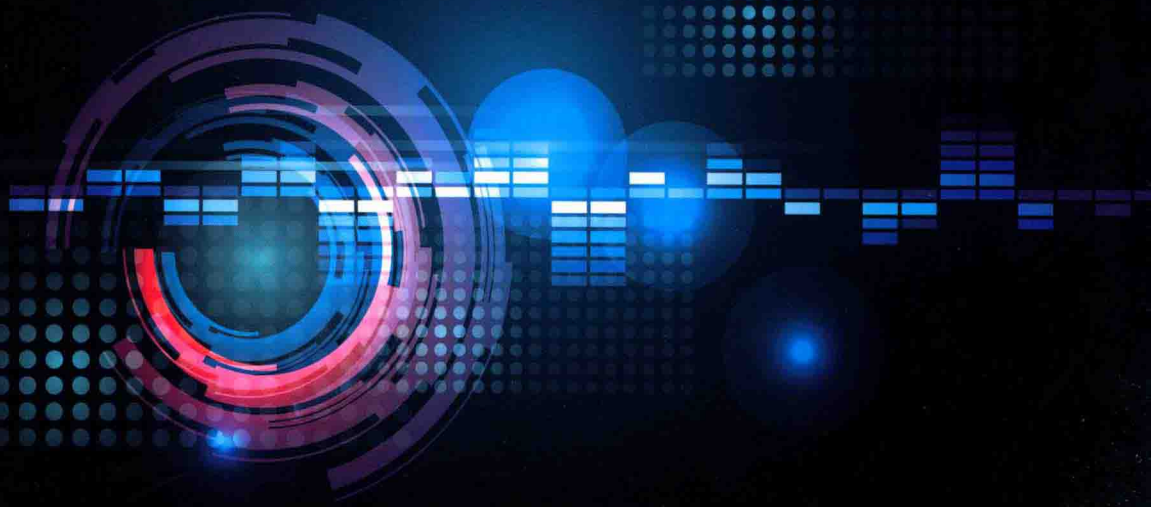
全国计算机等级考试二级Python优选教材

全国计算机等级考试二级
Python
优选教材



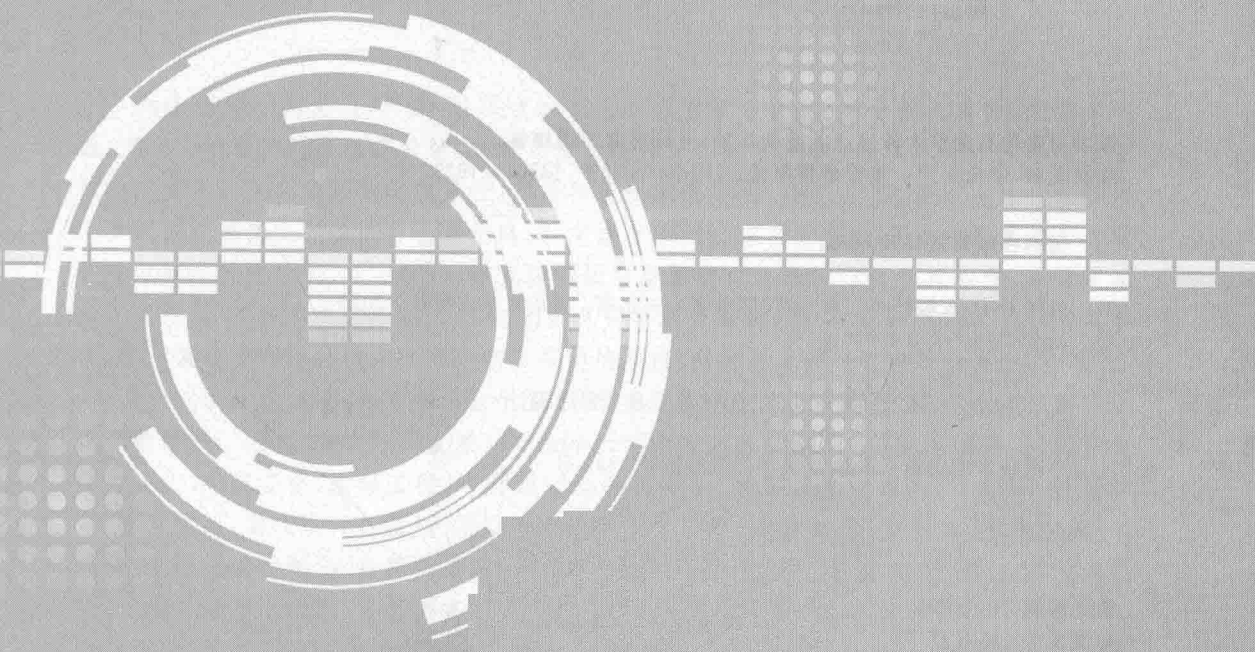
玩转Python

轻松过二级

董付国◎编著

- 用通俗易懂的语言和案例介绍Python基础知识、编程模式及应用，内容浅显易懂。
- 书中例题和演示代码配有大量注释，方便阅读和理解，大幅度缩短学习时间，能够快速掌握和运用。
- 结合二级Python考试大纲，提供730道课后练习题和参考答案，可以用来巩固和提高所学知识。
- 提供所有例题源代码，并支持多种方式与作者在线交流。

清华大学出版社



玩转Python

轻松过二级

董付国◎编著

清华大学出版社
北京

内 容 简 介

全书共12章，除了介绍Python编码规范，运算符、表达式与内置对象，Python序列结构，程序控制结构，函数，面向对象程序设计，字符串，正则表达式，文件内容操作，异常处理结构，以及SQLite数据库应用开发和tkinter编程精彩案例等知识之外，更重要的是提供了62道例题和730道练习题，尤其适合备考全国计算机等级考试二级Python模块的考生，以及需要大量练习题来巩固所学知识的Python爱好者。

本书全部代码适用于Python 3.5、Python 3.6以及更高版本。

图书在版编目(CIP)数据

玩转Python轻松过二级/董付国编著. —北京：清华大学出版社，2018
ISBN 978-7-302-49916-9

Ⅰ. ①玩…　Ⅱ. ①董…　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(2018)第055449号

责任编辑：白立军
封面设计：杨玉兰
责任校对：时翠兰
责任印制：丛怀宇

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：三河市铭诚印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：16.5　　**字　数**：380千字
版　　次：2018年6月第1版　　**印　次**：2018年6月第1次印刷
印　　数：1～2000
定　　价：49.00元

产品编号：078830-01

前言

FOREWORD

Python是一门免费、开源、跨平台的高级动态编程语言，支持命令式编程、函数式编程，完全支持面向对象程序设计，拥有大量功能强大的内置对象、标准库和涉及各行业领域的扩展库以及众多狂热的支持者，使得各领域的工程师、科研人员、策划人员甚至管理人员能够快速实现和验证自己的思路、创意或者推测。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中直接调用内置函数或标准库方法即可实现，大幅度减少了代码量，更加容易维护。Python用户只需要把主要精力放在业务逻辑的设计与实现上，在开发速度和运行效率之间达到了完美的平衡，其精妙之处令人击节叹赏。

Python由Guido van Rossum于1991年推出第一个公开发行版本，迅速得到各行业人士的青睐。经过20多年的发展，Python已经渗透到统计分析、移动终端开发、科学计算可视化、系统安全、逆向工程、软件测试与软件分析、图形图像处理、人工智能、机器学习、深度学习、游戏设计与策划、网站开发、数据爬取与大数据处理、密码学、系统运维、音乐编程、影视特效制作、计算机辅助教育、医药辅助设计、天文信息处理、化学、生物信息处理、神经科学与心理学、自然语言处理、电子电路设计、电子取证、树莓派等几乎所有专业和领域，在黑客领域更是多年来一直拥有霸主地位。与此同时，Python语言在各大编程语言排行榜上的位次也是逐年上升，在IEEE Spectrum 2017编程语言排行榜上名列榜首。

早在多年前Python就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、多伦多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。近几年来国内有几百所高等院校的多个专业陆续开设了与Python程序设计有关的课程，并且这个数量还在快速增加。浙江省已经确定2018年开始将高中信息技术课程中的Visual Basic替换为Python，并纳入高考，还有几个省市也正在探讨把Python纳入高考的可行性。同时，全国计算机等级考试二级也正式加入Python模块，并确定于2018年9月进行第一次考试。毫无疑问，这些措施是相当有意义的，极大促进了国内Python的普及和推广。

本书作者于1998年和1999年分别通过全国计算机等级考试二级Visual Foxpro、三级A和四级考试，并且取得四级优秀证书。2000年参加工作之后，又连续多年担任全国计算机等级考试监考老师和多家培训机构的二级C语言辅导班主讲教师，在这方面拥有丰富的经验。

本书作者自2011年开始潜心研究Python编程以及Python在各领域的应用，2015年开始陆续出版了《Python程序设计》《Python程序设计基础》《Python程序设计(第2版)》《Python可以这样学》《Python程序设计开发宝典》《中学生可以这样学Python》《Python

程序设计基础(第 2 版)》系列图书。近两年来作者应邀为多所高校、企业做报告和担任全国高校教师 Python 师资培训班主讲教师超过 30 次,并连续 7 个学期为不同专业学生讲授 Python 编程与应用,在 Python 教学方面积累了大量的经验。

内容组织与阅读建议

本书共 12 章,全部代码适用于 Python 3.5、Python 3.6 以及更高版本。

第 1 章　Python 概述。介绍 Python 编码规范、扩展库安装方法以及标准库对象和扩展库对象的导入。

第 2 章　运算符、表达式与内置对象。讲解 Python 运算符、表达式、常量与变量以及常用内置函数的用法。

第 3 章　详解 Python 序列结构。讲解列表、元组、字典、集合等序列结构的用法,以及列表推导式、生成器表达式、切片、序列解包等常用技术。

第 4 章　程序控制结构。讲解单分支选择结构、双分支选择结构、多分支选择结构和选择结构的嵌套,以及 for 循环和 while 循环的用法。

第 5 章　函数。讲解函数的定义与调用、递归函数,位置参数、默认值参数、关键参数和可变长度参数,以及 lambda 表达式的用法。

第 6 章　面向对象程序设计。介绍类与对象的概念,数据成员与成员方法的定义与使用,公有成员与私有成员的区别,封装与继承的概念,以及特殊方法的用法。

第 7 章　字符串。讲解字符串编码格式,字符串格式化,字符串常用方法,运算符与内置函数对字符串的操作,中英文分词,拼音处理等内容。

第 8 章　正则表达式。介绍正则表达式语法基础,Python 标准库 re 的常用函数。

第 9 章　文件内容操作。讲解内置函数 open()的用法,文件对象的常用方法,上下文管理语句 with,文本文件读写,二进制文件序列化和反序列化,以及 Word 文件和 Excel 文件操作。

第 10 章　异常处理结构。介绍异常的概念,以及常用的异常处理结构语法。

第 11 章　SQLite 数据库应用开发。介绍 SQLite 数据库基础,Python 标准库 sqlite3 的 Connection 对象与 Cursor 对象,常用 SQL 语句的语法,以及数据导入导出。

第 12 章　tkinter 编程精彩案例。介绍 Python 标准库 tkinter 常用组件,并通过大量实际案例演示这些组件的用法。

本书适用读者

本书可以作为(但不限于):

- Python 爱好者自学用书。
- 非计算机专业本科、专科程序设计课程教材。
- 全国计算机等级考试二级 Python 培训用书。
- 备考全国计算机等级考试二级 Python 考生的参考用书。
- 需要大量练习题来巩固和验证所学知识的 Python 爱好者。

致谢

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，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。当然，最应该感谢的是妻子和孩子对我这个代码狂人的理解和体谅。

感谢每一位读者，感谢您在茫茫书海中选择了本书，衷心祝愿您能够从本书中受益，学到真正需要的知识，祝每一位全国计算机等级考试二级 Python 考生都能取得优异的成绩！同时也期待每一位读者的热心反馈，随时欢迎您指出书中的不足，并通过微信公众号“Python 小屋”与作者沟通和交流！

本书的出版获山东省高水平应用型重点立项建设专业(群)项目资助，在编写出版过程中也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

2018 年 2 月

目 录

CONTENTS

第1章 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。目前最新版本分别是Python 2.7.14、Python 3.4.8、Python 3.5.5和Python 3.6.4，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，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。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。另外，Python 官方已宣布 2020 年开始不再维护 Python 2. x 系列。

1.3 Python 编程规范与代码优化建议

Python 非常重视代码的可读性，对代码布局和排版有非常严格的要求。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，但在定义函数时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。

(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。

(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。in 操作的时间复杂度对于列表和元组是线性的，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 按钮，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。

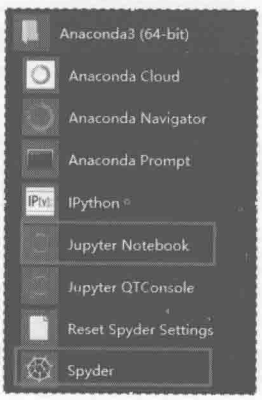


图 1-1　Anaconda3 菜单

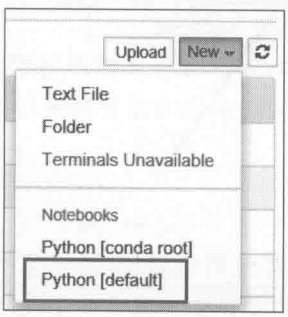


图 1-2　启动 Jupyter Notebook

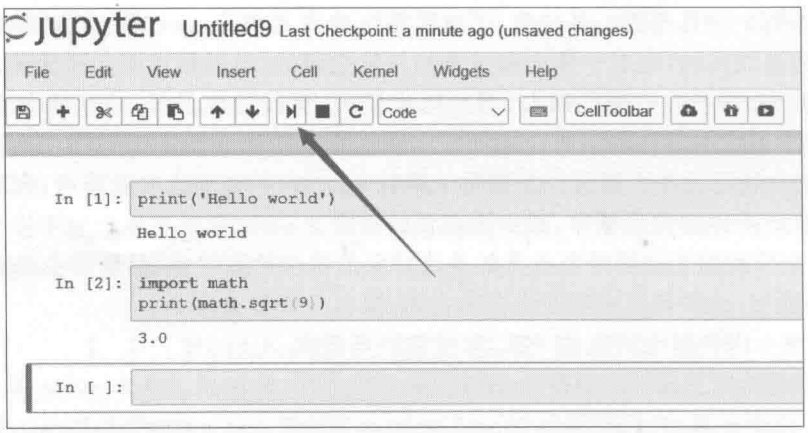


图 1-3 Jupyter Notebook 交互式编程界面

启动 Spyder 之后，可以选择使用主界面右侧的 IPython 或 Python 交互模式，也可以在主界面左侧编写程序文件并直接运行，如图 1-4 和图 1-5 所示。

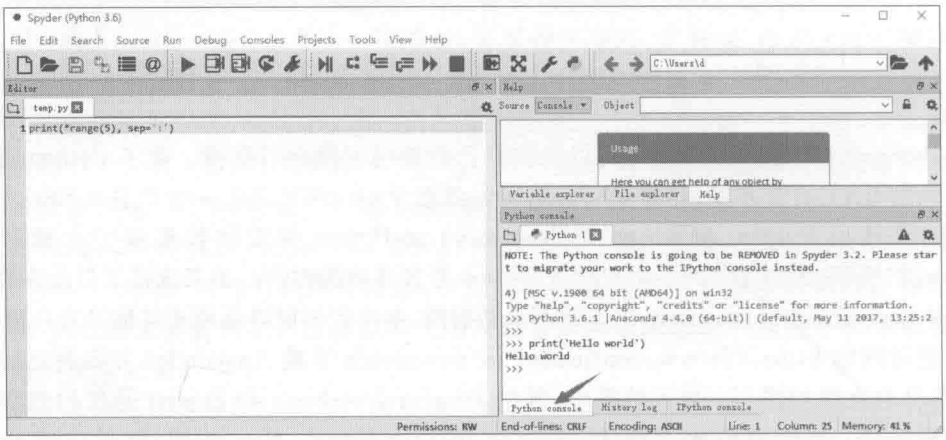


图 1-4 在 Spyder 中使用 Python 控制台交互编程环境

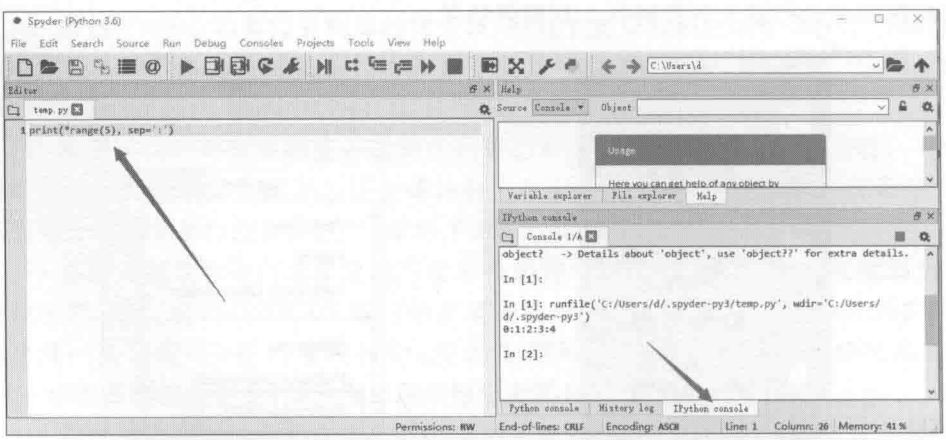


图 1-5 在 Spyder 中使用 IPython 交互编程环境，编写和运行程序文件

1.5　安装扩展库的几种方法

除了使用源码安装和二进制安装包(并不是所有扩展库都提供这种方式)以外,easy_install 和 pip 工具已经成为管理 Python 扩展库的主要方式,其中 pip 用得更多一些。使用 pip 不仅可以查看本机已安装的 Python 扩展库列表,还支持 Python 扩展库的安装、升级和卸载等操作。使用 pip 工具管理 Python 扩展库只需要在保证计算机联网的情况下输入几个命令即可完成,极大地方便了用户。常用 pip 命令的使用方法如表 1-1 所示。

表 1-1　常用 pip 命令的使用方法

pip 命令示例	说　明
pip download SomePackage[==version]	下载扩展库的指定版本,不安装
pip freeze	以 requirements 的格式列出已安装模块
pip list	列出当前已安装的所有模块
pip install SomePackage[==version]	在线安装 SomePackage 模块的指定版本
pip install SomePackage. whl	通过 whl 文件离线安装扩展库
pip install package1 package2…	依次(在线)安装 package1、package2 等扩展模块
pip install -r requirements. txt	安装 requirements. txt 文件中指定的扩展库
pip install --upgrade SomePackage	升级 SomePackage 模块
pip uninstall SomePackage[==version]	卸载 SomePackage 模块的指定版本

在 https://pypi. python. org/pypi 中可以获得一个 Python 扩展库的综合列表,可以根据需要下载源码进行安装或者使用 pip 工具进行在线安装,也有一些扩展库还提供了. whl 文件和. exe 文件,大幅度简化了扩展库的安装过程。有些扩展库安装时要求本机已安装相应版本的 C/C++ 编译器,或者有些扩展库暂时还没有与本机 Python 版本对应的官方版本,这时可以从 http://www. lfd. uci. edu/~gohlke/pythonlibs/下载对应的. whl 文件(注意,不要修改文件名),然后在命令提示符环境中使用 pip 命令进行安装。例如:

```
pip install pygame-1.9.2a0-cp35-none-win_amd64.whl
```

一般来讲,使用 pip 工具在线安装总是会自动选择扩展库的最新版本,但有时会出现新版本与其他扩展库不兼容的情况,或者其他扩展库依赖待安装扩展库的较低版本,这时可以明确指定扩展库的版本号,例如:

```
pip install requests==2.12.4
```

如果需要安装的扩展库比较多,并且对版本号要求严格,可以使用类似于 pip install -r requirements. txt 这样的命令从 requirements. txt 文件中读取所需安装的扩展库信息并自动安装。这个 requirements. txt 文件可以手工编辑,也可以使用 pip freeze >