普通高等教育“十三五”规划教材

（计算机专业群）

Python基础实例教程

秦颖　编著

中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十三五”规划教材（计算机专业群）

Python 基础实例教程

秦颖　编著

中国水利水电出版社
www.waterpub.com.cn

·北京·

内 容 提 要

Python 是近年来十分流行的编程语言。作为脚本语言，Python 尽管在速度上比编译语言如 C 和 C++等略有逊色，但其因开放性、跨平台和易学易用的特点获得了众多专业和非专业人士的青睐与支持。然而目前在介绍 Python 的书目中却难以觅到一本合适的教材，大部分资料为译著，内容过于宽泛，价格也不菲。所以编写一本适于初学者的实用学习教程，让读者把握 Python 的核心内容的实用教程成为我们本次编写的目的。

本书以凝练的风格介绍 Python 的核心知识，每一章都有明确的学习目标，并配有大量在交互环境下操练的实例和运行结果，以帮助读者理解具体的知识点。本书介绍了 Python 自带的开发环境以及 IPython 等其他集成开发环境，且全部实例的代码均在 Python 3 环境下调试通过。

全书共分 9 章，按照循序渐进的原则安排，从内置对象类型到语句语法，再到函数和模块，以及面向对象编程和异常处理等，较全面地覆盖了 Python 的基本内容，最后一章为典型程序代码和程序调试方法，为学习程序设计提供了样例。本书操作实例丰富实用，注重内容细节的介绍，对常用第三方模块也都有介绍。

本书适合作为高等院校计算机及相关专业的教材，适合 Python 初学者以及想快速了解 Python 语言特点的编程爱好者，也可为专业人士提供一定的参考。

图书在版编目（CIP）数据

Python基础实例教程 / 秦颖编著. -- 北京 : 中国水利水电出版社, 2019.2
普通高等教育“十三五”规划教材. 计算机专业群
ISBN 978-7-5170-7443-4

Ⅰ. ①P… Ⅱ. ①秦… 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031170号

策划编辑：周益丹　责任编辑：张玉玲　加工编辑：吕　慧　封面设计：李　佳

	普通高等教育“十三五”规划教材（计算机专业群）
书　名	Python 基础实例教程 Python JICHU SHILI JIAOCHENG
作　者	秦颖　编著
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（营销中心）、82562819（万水）
经　售	全国各地新华书店和相关出版物销售网点
排　版	北京万水电子信息有限公司
印　刷	三河市鑫金马印装有限公司
规　格	184mm×260mm　16 开本　11.75 印张　287 千字
版　次	2019 年 2 月第 1 版　2019 年 2 月第 1 次印刷
印　数	0001—2000 册
定　价	29.00 元

前　言

Python 语言诞生于 20 世纪 90 年代，迄今用户已达数百万。Python 是免费、开源的软件，简单易学却又功能强大，在目前主流操作系统平台上都能很好地运行 Python 脚本，这些特点使得 Python 获得了众多专业和非专业人士的青睐与支持，成为当前非常流行的一门编程语言，越来越多的行业都在应用 Python。从 YouTube 到大型网络游戏的开发，从动画设计到科学计算，从系统编程到原型开发，从数据库到网络脚本，从机器人系统到美国航空航天局（NASA）的数据加密，都有 Python 的用武之地。

Python 语言在当前的信息技术背景下获得了最佳的发展机遇，得到了迅猛发展。Python 已发展成为一种生态语言，第三方模块库已达到十几万个，并且还在不断丰富着。

本书重点介绍 Python 语言的核心基础知识，注重实践性。每一个知识点都先从理论角度分析，然后给出在交互环境下的操作实例，帮助读者加深对知识的理解，启发应用理论解决实际问题的思路。

本书对读者编程基础零要求，只要具备了计算机基础知识即可快速入门。Python 的交互模式提供了很好的语言学习环境，用户输入一条语句，语句马上能够执行，方便查看执行的结果。当然，集成开发环境 IDLE 也为大段脚本的编辑和调试提供了友好的环境。本书作为教材注重实用性，在力求简洁明确地说明知识点的同时，提供了多样而全面的操练题目，学生可以边操作边领悟，提高软件开发能力。本书既可作为计算机类专业学生的教材，也可作为 Python 应用开发者的参考书。

全书共分 9 章，内容安排循序渐进，由浅入深，层次清晰，通俗易懂。第 1 章介绍 Python 的特点和安装方法；第 2 章介绍 Python 内置对象类型，包括数字、列表、元组、字符串、字典、集合和文件等；第 3 章是 Python 的基本语句和语法，介绍了分支结构和循环结构语句的使用；第 4 章 Python 语言特有的一些内容，包括迭代、解析和生成器；第 5 章函数，介绍函数的定义和参数传递等关键问题；第 6 章模块，介绍模块的导入及变量的命名空间、几个常用 Python 标准库模块的使用方法；第 7 章面向对象程序设计初步，介绍 OOP 技术的核心概念以及在 Python 中实现 OOP 的基本方法；第 8 章介绍异常处理机制；第 9 章通过分析几个典型程序帮助读者快速上手编程，并对程序调试及排错给出一些建议和方法。

本书具有以下特点：

（1）语言简练，内容充实，较全面地覆盖了 Python 语言的核心内容。

（2）注重实用，不仅有理论分析，还精心设计安排了大量在交互环境下的实例，帮助理解知识点，提高动手能力，同时引领学生领悟 Python 语言的特点，提升应用 Python 语言解决问题的实践技能和创新意识。

（3）每一章都有内容总结和习题。习题丰富，形式多样，内容有趣味性，使学生能够享受到学习带来的乐趣和成就感。

（4）全面支持 Python 3，所有实例均在 Python 3 环境下进行了测试。

（5）教材提供配套的课件、部分习题的参考答案。

本书广泛收集和参考了各种 Python 的开源资料和文档，在出版过程得到了出版社的大力支持，在此向这些资料的分享者表示诚挚的感谢。

由于作者水平有限，书中难免有不妥和疏漏之处，恳请各位专家、读者批评指正，编者邮箱：qinying@bfsu.edu.cn。

编　者

2018 年 12 月

目　　录

第 1 章　认识 Python

本章将介绍 Python 语言的基本特点和应用、如何在各种平台上安装 Python，以及 Python 交互环境的使用，为下一步学习打下基础。

学习目标

- 认识基本概念：脚本语言、跨平台、交互环境。
- 掌握 Python 的基本特点、脚本语言的执行特点、Python 语言和其他语言的差异。
- 在个人计算机上完成 Python 的安装。
- 学习 Python 程序的各种执行方法，掌握 Python 交互环境的特点和使用方法。

1.1　Python 是什么

Python 是一门高级程序设计语言，是目前非常流行的开源脚本语言。据说 Python 之父 Guido van Rossum 给他发明的语言命名时，灵感源于一部 20 世纪 70 年代英国的喜剧连续剧 *Monty Python's Flying Circus*。Python 一词的英文含义是一种大型爬行类动物，只不过 Python 语言似乎和爬行动物并没有什么联系。

Python 主要是用 C 语言实现的，它的流行要归于它功能的强大。Python 可以在任何操作系统上运行，更重要的是，Python 是免费的开源软件，很多人在不断地完善着 Python 的功能。开发者们分享各个领域的应用，使 Python 越发强大，影响力越来越大。一般用户不仅可以免费下载安装 Python，还可以方便地共享第三方开发的免费功能模块。Python 的优良特性赢得了众多的拥护者和支持者，越来越多的行业开始应用 Python。从 YouTube 到大型网络游戏的开发，从动画设计到科学计算，从系统编程到原型开发，从数据库到网站开发，从机器人系统到美国航空航天局（NASA）的数据加密，都有 Python 的用武之地。尤其是在人工智能领域，Python 更是无可替代的编程语言。

除了标准的 Python 发布版本，还有众多的基于各种平台的变种，并提供了多样的语言开发环境。

（1）Enthought Python：同标准版的 Python 相比，Enthought Python 有一些花哨的工具和模块，很好用。安装 Enthought Python 将自动安装 IPython。IPython 提供了一个 Python 的交互式环境，但比默认的 Python 标准交互环境更友好。IPython 支持变量自动补全、自动缩进等操作，还内置了许多很有用的功能和函数，可以看作是 Python 交互的增强版。IPython Notebook 也称 Jupyter Notebook，它使用网络浏览器作为界面，进一步丰富了交互和可视化功能，非常适合作为教学工具。目前国外很多学校都以 IPython Notebook 为平台进行计算机相关课程的教学。

（2）ActivePython：一个适用于 Windows 平台的 Python 版本，内核是标准的 Python，由

Activestate 发布，包含了 Pythonwin 集成开发环境。

（3）以不同语言扩展实现的 Python：目前至少有 8 种，例如，PyPy 是用 Python 语言实现的 Python；Jython 是用 Java 语言实现的，在 Java 虚拟机上运行，使得 Python 脚本在本地机器上无缝连接到 Java 类库；IronPython 是用 C#实现的 Python，在 IronPython 中可以直接访问 C#的标准库。

归纳一下，Python 语言的主要特点有：

（1）易学。Python 学习入门很容易，即使没有编程基础的人，也可以在短时间内掌握 Python 的核心内容，写出不错的程序。因为 Python 的语句和自然语言很接近，所以十分适合作为教学语言。一个没有编程经历的人可以比较容易地阅读 Python 程序。下面来看一段用 Python 写的程序：

```
for line in open("file.txt"):
    for word in line.split():
        if word.endswith('ing'):
            print (word)
```

这段脚本实现的功能十分清晰，即：打开一个名为 file.txt 的文件，得到以空格分隔的一行中的单词，并把以 ing 结尾的单词都打印出来。简简单单的四行语句就完成了遍历和查找英文文本中现在分词或动名词的任务。可见，程序的易读性和简洁性是 Python 语言的第一大优点。

（2）跨平台性。软件的跨平台性又称为可移植性。Python 具有良好的跨平台性是指 Python 编写的程序可以在不做任何改动的情况下在所有主流计算机的操作系统上运行。换句话说，在 Linux 下开发的一个 Python 程序，如果需要在 Windows 系统下执行，只要简单地把代码拷贝过来，在安装了 Python 解释器的 Windows 计算机上就可以很流畅地运行，不需要做任何改动。跨平台性正是各种平台的用户都喜欢 Python 的重要原因之一。

（3）强大的标准库和第三方软件的支持。Python 中内置了大约 200 个标准功能模块，每一个模块中都自带了强大的标准操作，用户只要了解功能模块的使用语法，就可以将模块导入到自己的程序中，使用其标准化的功能，实现积木式任务开发，极大地提高程序设计的效率。导入模块的本质是加载一个别人设计的 Python 程序，并执行那个程序的部分或全部功能。除了 Python 标准库模块外，还有大量第三方提供的功能模块，如 Pyinstaller、Numpy、Scipy、Pandas、Matplotlib 等也都是免费的，并且得到了广泛使用，极大地丰富和增强了 Python 的功能。

（4）面向对象的脚本语言。脚本（Script）语言是与编译（Compile）语言不同的一种语言。脚本程序的执行需要解释器，且具有边解释边执行的特点。编译语言编写的程序需要把全部语句编译通过后才能执行。典型的编译语言有 C 和 C++。脚本语言和编译语言相比，通常语法比较简单，但是语言简单不等同于只能用于简单任务的编程。相反，Python 的简单和灵活使得很多领域的复杂任务开发变得十分容易。在本书中，我们也经常将 Python 程序称为脚本。同时，Python 也是一种面向对象程序设计语言，它具有完整的面向对象程序设计的特征，如 Python 的类对象支持多态、操作符重载和多重继承等面向对象的特征，因此用 Python 实现面向对象程序设计十分方便。和 C++和 Java 等相比，Python 甚至是更理想的面向对象设计语言。

1.2　Python 的安装

作为一种开源软件，Python 的使用和发布都是免费的，用户可以访问 Python 的官方网站（http://www.python.org/download/）来获取最新版本的 Python 安装程序。本书截稿时 Python 最新的版本为 3.7，陆续还将有新的版本推出。需要注意的是，不同平台的安装版本不同，要根据相应的平台选择不同的版本下载。在常见的操作系统如 Windows、Linux、UNIX 和 Macintosh（Mac）上都可以顺利地安装 Python 的解释器。通常 Linux、UNIX 和 Mac OX 系统中都包含了 Python 的某个版本，因此不需要单独安装。安装之前先查看一下自己系统中是否已经安装了 Python 解释器。Linux 和 UNIX 系统中 Python 一般安装在/usr 路径下。对于 Windows 系统的用户，需要自行安装 Python。安装成功后可以在菜单“开始”→“所有程序”中看到 Python。下面详细介绍在 Windows 和其他操作系统中 Python 的具体安装步骤。

1.2.1　Windows 平台

在 Python 官方网站下载能够在 Windows 下运行的.msi 安装程序。安装程序又分为适用于 32 位机的Windows x86 MSI installer和适用于 64 位机的Windows x86-64 MSI installer两个版本，读者需要根据自己操作系统的位数做出正确选择，否则将无法正常运行。图 1-1 所示是运行 Python 3.2 安装程序的界面。Python 解释器的默认安装路径为 C:\python32。运行 Python 安装程序，第一步是确定安装路径。

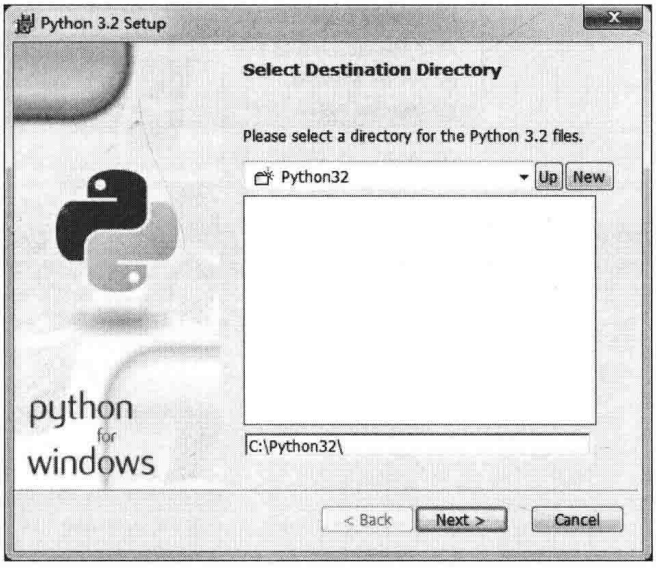

图 1-1　Python 3.2 安装程序的界面

下一步是定制安装内容。默认安装的有 Python 解释器、标准库和说明文档等内容。可以点开每一项左侧的黑箭头来改变默认设置，增减安装内容，如图 1-2 所示。

安装过程根据向导一步步地进行即可。成功安装后，从“开始”菜单就能看到 Python 了，如图 1-3 所示。

其中 IDLE 为 Python 自带的图形界面集成开发环境，用于 Python 程序的设计和调试。IDLE

的图形窗口如图 1-4 所示，是一个可以交互式地输入语句的环境，也支持基本的编辑操作，如复制和粘贴等。对于已经执行过的语句，按 Alt+P 组合键可以上翻，按 Alt+N 组合键可以下翻，以避免重复录入。

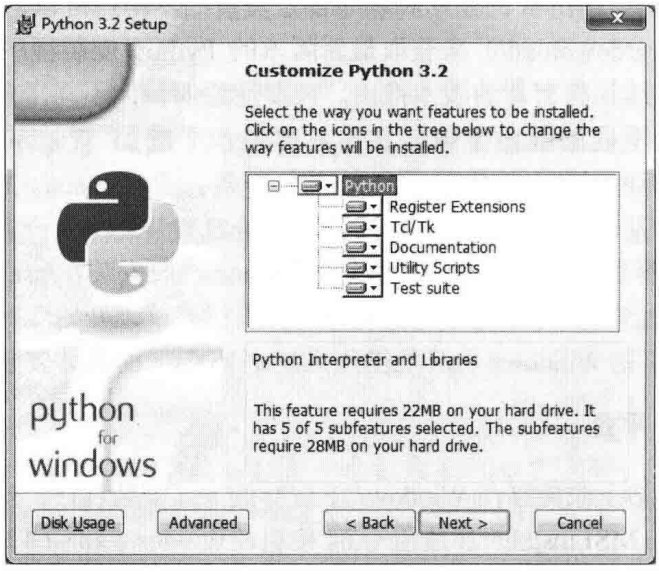

图 1-2　定制安装内容

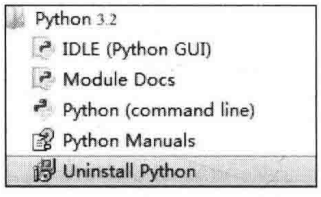

图 1-3　“开始”菜单

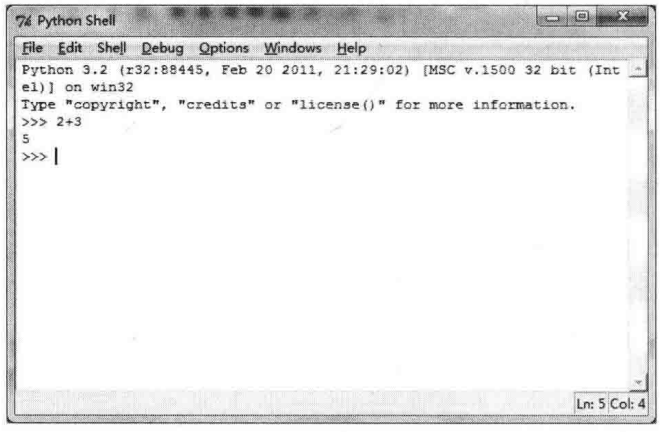

图 1-4　IDLE 图形窗口

如果要编辑大段的脚本，要单击交互环境中的 File→New Window 菜单命令新建一个窗口来完成大段脚本的输入，如图 1-5 和图 1-6 所示，然后用 File→Save 菜单命令保存为*.py 文件。新窗口中的编辑菜单 Edit 提供了多种常规的编辑操作命令。在 IDLE 环境下编辑脚本，不同数据

类型、内置函数、语句等都会以不同的颜色显示，以帮助编程者及时发现脚本输入过程中的语法错误。比如，字符串常量是以引号括起来的，输入引号后，后面的内容自动变为绿色，表示字符串常量；Python 内置函数输入正确时为紫色，关键词等为橙色显示。当然颜色体系是可以进行设置的。

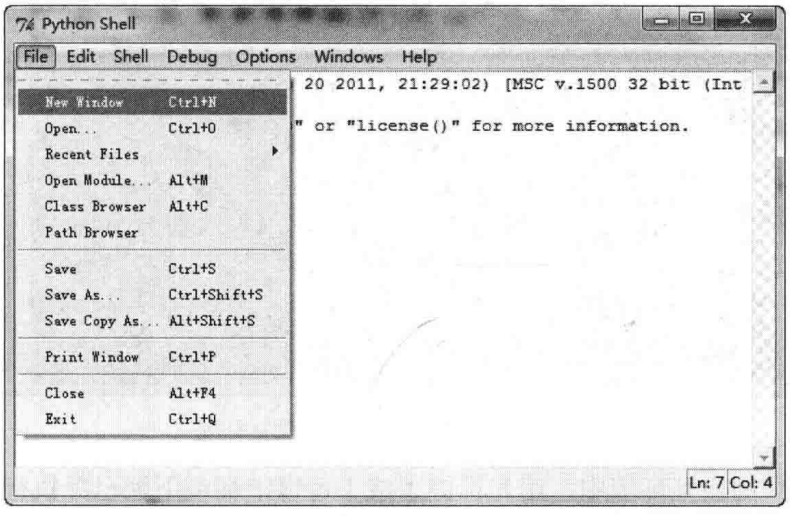

图 1-5　执行菜单命令

图 1-6　新建窗口

Command line 也提供 Python 的交互模式，是在命令行窗口运行的交互环境，如图 1-7 所示。进入该交互环境后，提示符为>>>。在提示符后可输入 Python 的表达式或语句。Python 的交互环境主要用于简单程序的交互执行和代码的验证、测试。输入一条语句或表达

式后立即执行，并在下一行显示结果（如果有输出结果的话）。

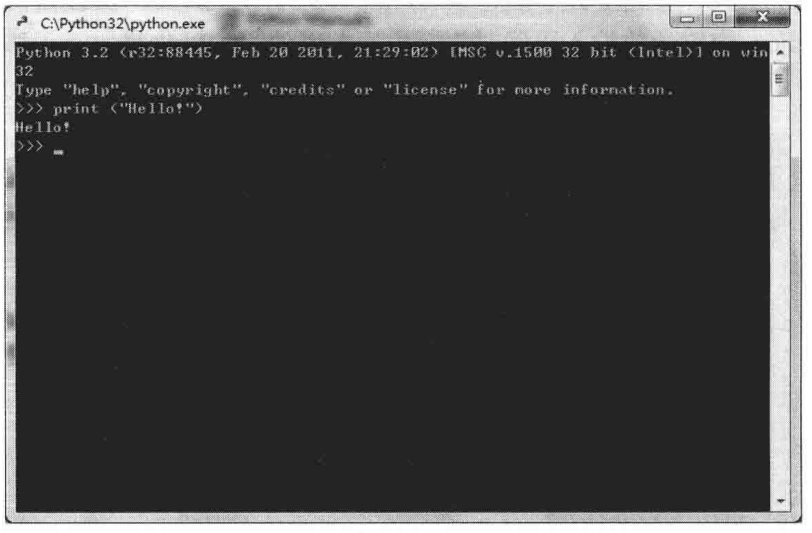

图 1-7　命令行窗口运行的交互环境

在命令行窗口的交互环境中，用光标键↑或↓可以上翻或下翻已经执行过的命令，以提高输入效率。

比自带的开发环境更好用的其他开发环境有很多，如 Anaconda、Pycharm、Wing 等。其中 Anaconda 是免费的开发环境，下载地址为 https://www.continuum.io。Anaconda 中不仅集成了调试工具 Spyder，还集成了交互编程环境 IPython，拥有众多用户。

Modules Docs 和 Manuals 是 Python 的文档和标准手册，是可供随时查阅的文档。从“开始”菜单中选择 Python 安装文件夹下的 Python Manuals 即可打开帮助窗口，如图 1-8 所示。

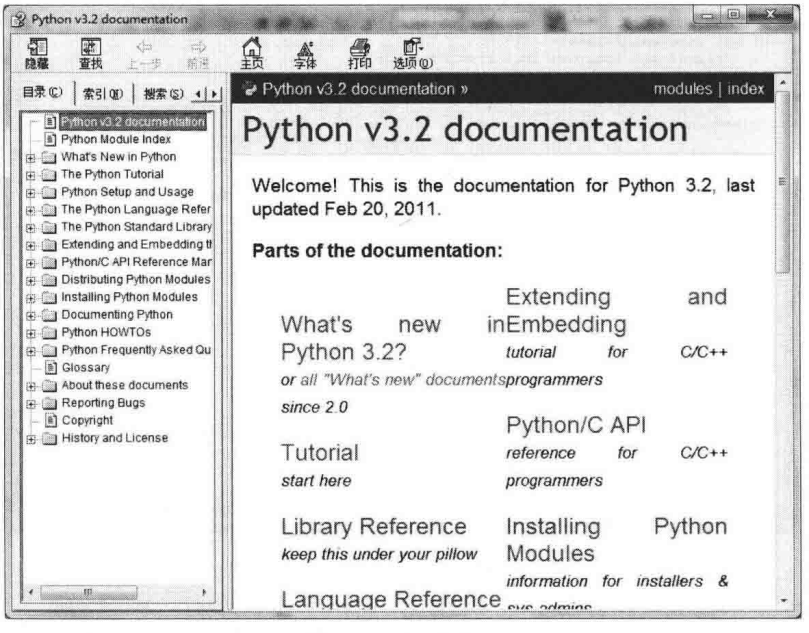

图 1-8　帮助窗口

在交互环境下同样可以使用 dir 和 help 来获得关于 Python 的函数、对象属性、方法等有

关的帮助信息，具体做法如图 1-9 所示。

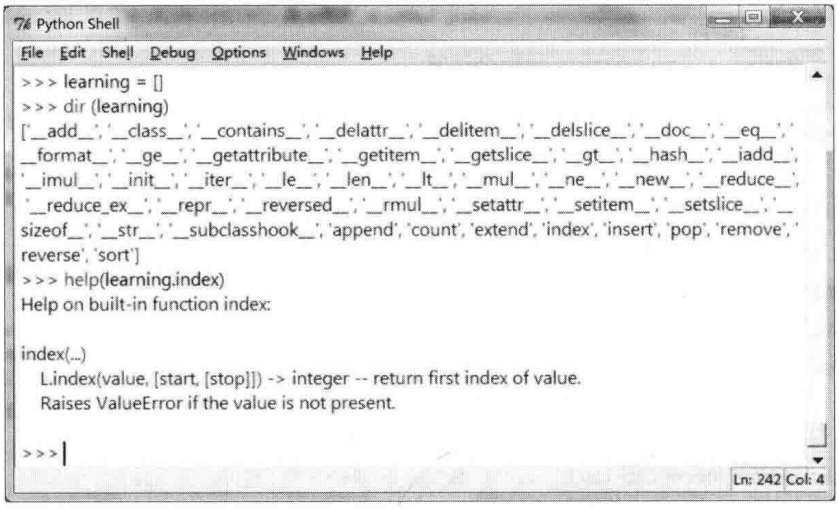

图 1-9　使用 dir 和 help 获得帮助

1.2.2　Linux、UNIX 和 Macintosh

在 Linux 下 Python 安装文件是一个或多个 rpm 压缩文件。下载解压缩后运行 config 和 make 命令，Python 可自动完成系统配置，也可以参照压缩包中的 readme 文件给出的步骤完成安装。有的针对 Linux 平台的 Python 为自解压安装文件，可以通过执行下面的命令自行安装，比如 Canopy 的安装。

```
bash canopy-1.5.3-rh5-64.sh
```

安装成功后，在 shell 提示符（终端窗口）后输入 python 就会出现 Python 的提示符>>>。

1.3　执行 Python 程序的方法

运行 Python 脚本有多种方式，如交互环境下运行、命令行窗口运行。Windows 和 Linux 平台还有各自的特点。

1. IDLE 环境运行

打开集成开发环境 IDLE，默认进入交互环境，出现提示符>>>。在 >>> 提示符后输入 Python 的语句或表达式，语句和表达式都会立即被执行。一次执行一条，语句正确时就显示表达式或语句运行的结果，方便计算表达式的值。有人甚至把 Python 的交互环境当作一个计算器来使用。交互环境下最后一个表达式的结果还被保存在一个特殊的变量“_”中，利用“_”变量构成新的表达式就可以直接在前面运算结果的基础上继续运算。交互环境下不仅可以使用常量，也可以定义变量。下面是交互环境下的操作实例（#后为注释部分，不执行）。

```
>>> 2+5
7
>>> (50 - 5.0*6)/4
5.0
>>> 0.1+0.2
```

```
0.30000000000000004          #请思考为什么
>>>17 // 5.0                 #floor 除法
3.0
>>>17 % 3                    #取余运算
2
>>> 3**2                     #指数运算
9
>>>50+_                      #特殊变量“_”保存最后一次的计算结果
59
>>> width = 20
>>> height = 5 * 9
>>> width * height
900
>>> X
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'X' is not defined
```

Python 中使用变量前必须先定义变量。上面最后一个例子中，我们输入了一个没有定义的变量 X，解释器无法找到 X 的定义，因此提示名字错误。在交互环境下，即使输入的表达式或语句存在语法错误，Python 解释器也不会崩溃，而是有相应的错误提示信息，描述错误的位置和内容。换句话说，Python 在交互环境下具备异常处理功能。关于异常处理的知识将在第 8 章介绍。

注意：在交互模式下，一次只能执行一条语句，而且输入的代码都不会被保存下来，关闭解释器时将全部消失。如果需要编写较长的脚本，应该利用文本编辑器编写，也可以利用集成开发环境 IDLE 来编辑脚本。选择 file→New Window 菜单命令新建编辑窗口。代码以.py 为扩展名保存，有的 IDLE 不提供默认的扩展名。

在交互模式下如果输入的是复合语句（目前可简单地认为复合语句是以“:”结尾的语句），在第一行内容输入结束按 Enter 键后，提示符会自动变为“...”或自动缩进一定的距离，提示输入复合语句的剩余部分。需要特别注意的是，语句的缩进在 Python 脚本中相当重要，不同的缩进量反映了语句块的不同层次和关系。当结束复合语句的输入时，按两次 Enter 键就开始执行语句。例如下面的 for 循环语句是一个复合语句，其后缩进的语句块都是 for 循环体内的语句。

```
>>> for i in 'python':
    ...    print (i)
    ...
                                #结束输入，运行
p
y
t
h
o
n
```

Python 3.x 后全面支持 Unicode 字符，因此字符串中可以包含中文。请看下面的输出结果。

```
>>> st="中英文 Hello"
```

```
>>> for ss in st:
        print(ss)

中
英
文
H
e
l
l
o
```

IDLE 环境下执行一段程序时，首先要打开脚本，然后选择 run→run Module 菜单命令或按 F5 功能键。

2. 命令行窗口运行

在 Windows 系统中，执行“开始”→“运行”命令，在弹出的对话框中输入 cmd，打开命令行窗口。假设已经编辑完成了一段名为 myscript.py 的 Python 脚本文件，要运行 myscript.py 可以在命令行中输入：

```
C:\python32> python mysrcipt.py
```

Python 解释器加脚本文件是最简单的在命令行窗口中运行 Python 程序的方式。完整的命令行运行命令格式有些复杂，形式如下：

```
python [-BdEiOQsRStuUvVWxX3?] [-c command | -m module-name | script | - ] [args]
```

该命令中有多个参数，方括号括起来的参数是可选参数，如果所有参数都缺省，则只运行 Python 解释器，从而进入交互环境。第一个方括号内为多种选项参数，这里不一一介绍了；第二个参数-c 后为若干命令，是要求 Python 语句以指定的命令方式执行；第三个方括号内参数-m 后跟模块名，即将指定的模块作为程序的主模块，当程序由多个模块构成时这个参数才有用；最后方括号内的参数 args 为通用选项，常用的如-h 为帮助信息，-V 为显示当前 Python 的版本号。

3. Windows 环境下运行

在 Windows 平台一般通过双击 Python 脚本文件的图标是可以运行 Python 程序的，不过这样做会遇到一些问题，比如一些程序的运行结果（甚至包括错误提示）会一闪而过，不能像命令行窗口运行程序那样将结果停留在屏幕上。如果想看到程序的运行结果，一种解决方法就是打开脚本文件，在程序最后添加一条 input()语句，将文件保存后，再次双击运行程序，这样就可看到运行结果或错误提示了。因为当程序执行到最后的 input()时，会等待用户输入任意内容再关闭窗口。

在 IDLE 中，不仅提供编辑脚本的环境，同时还提供代码调试的功能。执行 Run→Run Modules 菜单命令就可执行正在编辑的脚本。运行结果和异常等都出现在交互环境下。这也算是一种窗口环境下运行 Python 脚本的方式。

4. Linux 和 UNIX 环境下运行

在 Linux 和 UNIX 环境下，用 chmod 命令设置脚本文件为可执行属性（+x），并在第一行代码中说明 Python 解释器的路径，就可以输入脚本文件名直接运行程序。假设 Python 解释器安装在/usr/local/bin/python 下，在脚本文件第一行添加这样一行信息：

```
#! /usr/local/bin/python
```

即在脚本文件的第一行说明 Python 解释器所在的位置。然后修改脚本文件的属性为可执行，之后直接输入如下脚本程序文件名即可运行程序：

```
% myscript
```

1.4 交互环境 IPython/Jupyter

IPython 是一个强大的交互式计算环境。Python 的大部分功能都可以在 IPython 交互环境下完成。虽然 Python 自带的 IDLE 也是一种交互环境，但 IPython 提供了更丰富的功能，内置了许多很有用的函数，如魔术（magic）函数等，实现了交互式的数据可视化，并提供了高效的并行计算功能。IPython Notebook 则是基于 Web 的 C/S 技术开发的交互式计算文档格式，将 Python 的代码、图形、输入以及描述合并到一个文档中。目前已有很多学校将 IPython Notebook 作为计算机和其他专业的教学环境，成为 Python 编程、机器学习、数据分析等课程的教学工具，十分受欢迎。IPython Notebook 也称 Jupyter Notebook，因为 IPython 的内核为 Jupyter，可以在 Jupyter Notebook 和其他交互前端环境下使用 Python 语言。Notebook 文件的扩展名为.ipynb，是一个 json 格式的文件，包含了在交互会话中的输入输出。

1. 安装 IPython

如果系统已经安装了 pip，则可以用 pip 快速安装 IPython，安装命令如下：

```
pip install ipython
```

如果想使用 IPython Notebook 提供的 Web 界面，还需要安装内核 Jupyter，安装命令如下：

```
pip install jupyter
```

成功安装后，启动 IPython 的界面出现如下提示：

```
Python 3.6.0
Type 'copyright', 'credits' or 'license' for more information
IPython 6.0.0.dev -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

注意到 IPython 的提示符不是 Python 通常的>>>，而是“In[N]:”。在 In[1]:后面输入语句或表达式，按 Enter 即可执行。语句 N 运行的输出结果显示在“Out[N]:”后面，如下：

```
In [1]:print ("Hello, IPython!")
Hello, IPython!
In [2]: 21 * 2
Out[2]: 42
```

IPython 提供了 Tab 键代码自动补全、自动缩进功能，还有以%开头的“魔术命令”。用%magic 命令就可以显示所有魔术命令的详细文档。常用的魔术命令有：

- %quickref：快速查询 magic 命令。
- %run script.py：在 IPython 中运行脚本文件 script.py。
- %time statement：测试 statement 的运行时间。
- %hist：显示所有会话中输入命令的历史记录。

更多关于 IPython 的使用请自行参考 IPython 的相关文档，网址为 http://ipython.readthedocs.io/en/stable/index.html。