

高等院校信息技术规划教材

Python实用教程

秦颖 编著



清华大学出版社

高等院校信息技术规划教材

Python实用教程

秦颖 编著

清华大学出版社
北京

内 容 简 介

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和支持。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。一本适合初学者,既能让读者把握 Python 的核心内容而又不失简洁性,便于理解和操作的实用学习教程正是本书的目标。

本书介绍 Python 核心知识,每章都有明确的学习目标,配有大量在交互环境下的操练实例和运行结果以帮助读者理解知识点。全书共 9 章,按照循序渐进的原则安排内容,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 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。

本书适合作为高等院校计算机及相关专业本科生的教材,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。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Python 实用教程/秦颖编著. —北京:清华大学出版社,2016
高等院校信息技术规划教材
ISBN 978-7-302-45030-6

I. ①P… II. ①秦… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2016)第 218489 号

责任编辑:张 玥 赵晓宁

封面设计:常雪影

责任校对:徐俊伟

责任印制:何 芊

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:三河市吉祥印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:11.75

字 数:279 千字

版 次:2016 年 11 月第 1 版

印 次:2016 年 11 月第 1 次印刷

印 数:1~2000

定 价:29.50 元

产品编号:071045-01

Python 语言诞生于 20 世纪 90 年代,迄今为止 Python 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运行 Python 脚本,这些特点使得 Python 得到了众多的专业和非专业人士的青睐和支持,成为当前十分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国家宇航局 NASA 的数据加密,都有 Python 的用武之地。

本书重点介绍 Python 语言的核心基础知识,注重实践性。书中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。本书为北京外国语大学中国外语教学研究中心资助项目。

本书对读者编程零基础要求,只要具备了计算机导论知识的读者都可以快速入门。Python 的交互模式提供了很好的语言学习环境,用户可以输入一条语句,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,还提供了多样而全面的操练题目,学生可以边操作边领悟,提升软件开发能力。本书既可以作为计算机类专业各层次学生教材,也可以作为 Python 应用开发者的参考用书。

全书共 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;第 2 章介绍 Python 内置数据类型,包括数值、字符串、列表、元组、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 章为 Python 语言特有的一些内容,包括迭代、解析和生成器;第 5 章是函数,介绍函数的定义和参数传递等关键问题;第 6 章为模块,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 章为面向对象程序设计 OOP,介绍 OOP 技术的核心概念以及用 Python 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,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,帮助读者理解知识点,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。习题形式多样,富有趣味性,使得学生能享受学习带来的乐趣和成就感。

(4) 提供配套的课件以及部分习题的参考代码。

本书广泛收集和参考了各种 Python 的开源资料和文档,在这里对这些资料的作者致谢。

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、同仁和读者不吝赐教,并与笔者讨论。邮箱 qinying@bfsu.edu.cn。

编者

2016年10月

第 1 章 认识 Python	1
1.1 Python 概述	1
1.2 Python 的安装	3
1.2.1 Windows 平台	3
1.2.2 Linux、UNIX 和 Macintosh	7
1.3 执行 Python 程序的方法	7
小结	10
习题	10
第 2 章 Python 内置对象类型	12
2.1 数字	12
2.1.1 数字常量	13
2.1.2 表达式操作符	14
2.1.3 数字的其他类型	15
2.2 列表和元组	16
2.2.1 列表	16
2.2.2 通用序列操作	18
2.2.3 列表的基本操作	21
2.2.4 列表对象的基本方法	23
2.2.5 列表的应用	27
2.2.6 元组	28
2.3 字符串	29
2.3.1 字符串常量	29
2.3.2 基本字符串的操作	31
2.3.3 字符串的格式化	37
2.3.4 转换字符串	40

2.4	字典	41
2.4.1	字典的定义和构建	41
2.4.2	字典的基本操作	43
2.5	集合	46
2.5.1	集合的特点	46
2.5.2	集合的运算	47
2.5.3	集合对象的方法	48
2.6	文件	49
2.6.1	文件的读写操作	49
2.6.2	二进制文件和文本文件	52
	小结	54
	习题	54
第 3 章	语句和语法	58
3.1	赋值语句	59
3.1.1	赋值语句和变量命名	60
3.1.2	赋值的形式	61
3.2	if 语句	64
3.2.1	if 语句的格式	64
3.2.2	多行语句	65
3.2.3	测试条件的形成	66
3.2.4	if/else 表达式	68
3.2.5	嵌套 if 结构	68
3.3	while 和 for 循环语句	69
3.3.1	while 循环	69
3.3.2	for 循环	72
3.3.3	与循环有关的内置函数	74
	小结	76
	习题	77
第 4 章	迭代、解析和生成器	80
4.1	迭代	80
4.2	解析	82
4.2.1	列表解析	82
4.2.2	字典和集合解析	83
4.3	生成器	84
4.3.1	生成器函数	84

4.3.2 生成器表达式	86
小结	87
习题	87
第 5 章 函数	89
5.1 函数的定义和调用	89
5.2 参数传递	91
5.2.1 参数传递的两种模式	91
5.2.2 参数的匹配	92
5.3 变量的作用域	96
5.4 递归	99
5.5 匿名函数 lambda	100
5.6 一个函数实例	102
小结	103
习题	103
第 6 章 模块	106
6.1 模块导入	106
6.2 标准库模块	109
6.2.1 sys	109
6.2.2 os	110
6.2.3 fileinput	113
6.2.4 random	114
6.2.5 re	115
6.2.6 getopt	121
6.2.7 time	123
6.3 模块的搜索路径	124
6.4 创建模块	125
6.5 主模块	126
小结	127
习题	127
第 7 章 面向对象程序设计初步	132
7.1 面向对象基础	132
7.2 类和实例	133
7.2.1 类和实例的生成	133
7.2.2 类的继承	135

7.3 类的设计	137
7.3.1 构造函数	137
7.3.2 类方法的设计	138
7.3.3 运算符重载	140
小结	142
习题	143
第 8 章 异常基础	145
8.1 触发异常和捕获异常	145
8.1.1 触发异常	146
8.1.2 捕捉异常	147
8.2 用户定义的异常类	151
8.3 with/as 环境管理器	152
小结	153
习题	153
第 9 章 程序实例和调试	155
9.1 英文单词词形还原	155
9.2 嵌套的同音单词	156
9.3 网络爬虫	158
9.4 程序调试	163
9.4.1 语法错误	163
9.4.2 运行时错误	163
9.4.3 语义错误	164
附录 A	165
A1 Python 2.7. x 和 Python 3. x 的主要差别	165
A2 Python 中的保留字	169
A3 Python 内置异常	169
参考文献	177

认识 Python

本章介绍 Python 语言的基本特点和应用,内容包括在各种平台上如何安装 Python 以及 Python 的交互环境的使用,是下一步学习的基础。

学习目标

- 认识脚本语言、跨平台、交互环境等基本概念;
- 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 PC 上完成 Python 的安装;
- 学习 Python 程序的各种执行方法,掌握 Python 交互环境的使用。

1.1 Python 概述

Python 是一门高级程序设计语言,也是目前十分流行的开源脚本语言。据说 Python 之父 Guido van Rossum 为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。而 Python 英文一词的含义是一种大型爬行类动物,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,它的流行要归功于它的功能强大。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,影响力越来越强。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,越来越多的行业中都在应用 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国家宇航局 NASA 的数据加密,都有 Python 的用武之地。

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,下面简单介绍几种。

- 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 也提供了一个 Python 的交互式环境,但比默认的 Python 标准交互环境友好。

ipython 支持变量自动补全、自动缩进等操作,内置了许多很有用的功能和函数,可以看做是 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,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。

- 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。
- 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;Jython 是用 Java 语言实现的,可在 Java 虚拟机上运行,使得 Python 脚本在本地机器上可无缝链接到 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。

综上,Python 的主要特点有以下几个。

(1) 易学。Python 入门容易,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。一个没有编程经验的人也可以比较容易地读懂 Python 程序。让我们来看一段用 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,就是: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 4 行语句就完成了遍历和查找英文文本文件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。软件的跨平台又称为可移植性。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,只要简单地把代码复制过来,在安装了 Python 解释器的 Windows 计算机上就可以很流畅地运行,而不需要做任何改动。跨平台性正是各种平台的用户都喜欢 Python 的重要原因。

(3) 强大的标准库和第三方软件支持。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格式,就可以将模块导入到自己的程序中,使用其中标准化的功能,实现积木式任务开发,极大地提高了程序设计的效率。导入模块的本质是加载一个别人设计的 Python 程序,并执行该程序的部分或全部功能。除了 Python 标准库模块外,还有大量的第三方提供的功能模块,如 Numpy、Scipy 等都是免费的,得到了广泛的使用,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)语言相对的一种语言。脚本程序的执行需要解释器。脚本程序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。典型的编译语言是 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高效。在本书中,也经常将 Python 程序称为脚本。

同时 Python 是一种面向对象程序设计的语言,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,因此用 Python 实现面向对象程序设计十分方便。与 C++ 和 Java 等相比,Python 甚至是更理想的面向对象的设计语言。

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,用户可以从 Python 的官方网站 <http://www.python.org/download/> 上,方便地获取最新版本的 Python 安装程序。目前 Python 最新的版本为 3.5.1,还将有新版本推出。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 和 UNIX 以及 Mac OS 系统中都包含了 Python 的某个版本,一般不需要单独安装。因此,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 Windows 系统的用户,Python 没有在系统中,需要用户自行安装。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 和其他操作系统中的具体安装步骤。

1.2.1 Windows 平台

在 Python 官网下载能够在 Windows 下运行的 .msi 安装程序。安装程序又分为适用 32 位系统的 Windows x86 MSI installer 和 64 位系统的 Windows x86-64 MSI installer 两个版本,读者需要根据自己的操作系统位数做出正确的选择;否则将无法正常运行。图 1-1 是运行 Python 3.2 安装程序的界面。Python 解释器默认安装路径为 C:\python32。运行 Python 安装程序首先是确定安装路径。

然后是定制安装内容。默认安装的有 Python 解释器和标准库以及说明文档等内容。可以单击每一项左侧的下三角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 具有图形界面的集成开发环境,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。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 键可以上翻,按 Alt+N 键可以下翻。

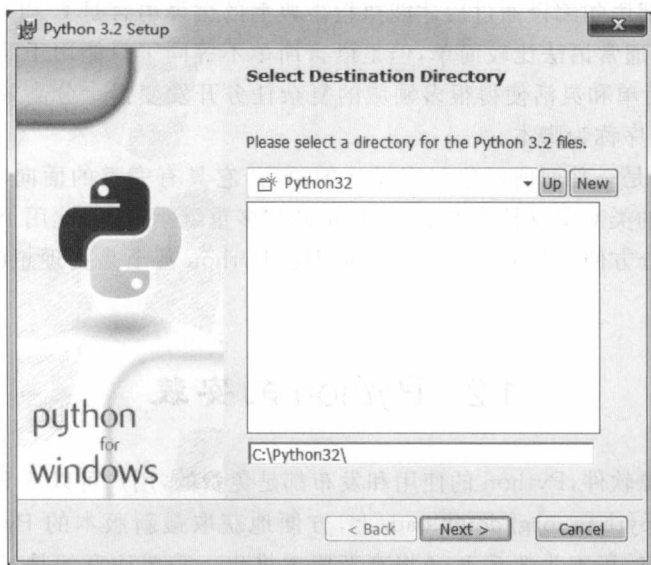


图 1-1 Python 3.2 安装程序界面

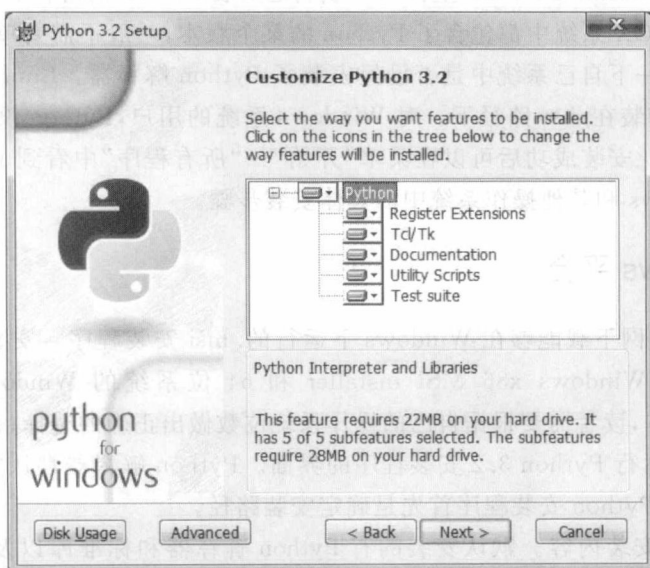


图 1-2 定制安装内容

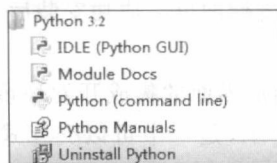


图 1-3 “开始”菜单

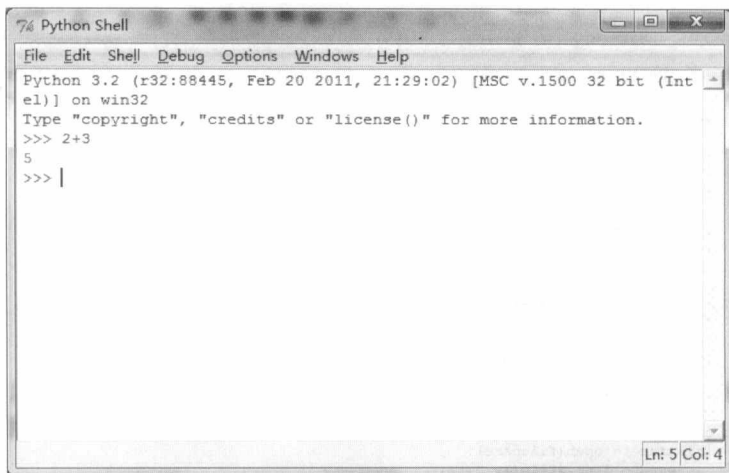


图 1-4 IDLE 图形窗口

如果要编辑大段的脚本,要用交互环境中的 File→New Window 菜单命令新建一个窗口来完成脚本的输入,如图 1-5 和图 1-6 所示,然后选择 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会自动变为绿色,表示字符串常量;而 Python 内置函数输入正确时为紫色,关键词等为橙色显示。当然颜色体系也是可以重新设置的。

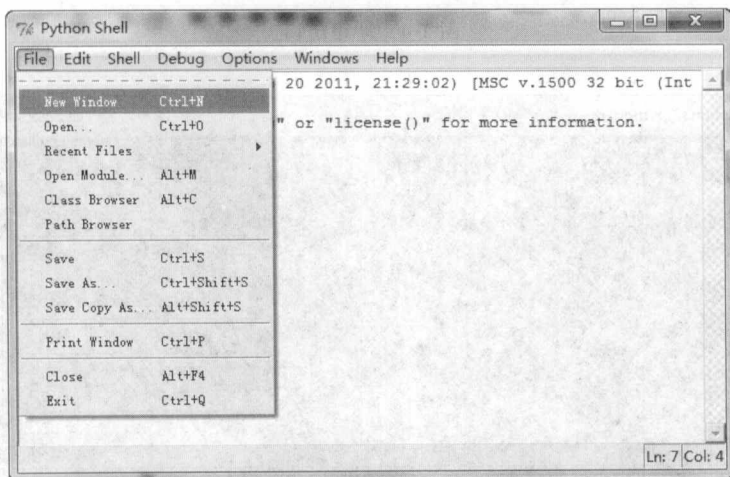


图 1-5 执行菜单命令

Command Line 也是 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 表达式或语句。Python 中的交互环境主要用于简单程序的交互执行和代码的验证及测试。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)。



```
7% anagram_sets.py - C:\Python27\mylab\anagram_sets.py
File Edit Format Run Options Windows Help

def signature(s):
    """Returns the signature of this string, which is a string
    that contains all of the letters in order.
    """
    t = list(s)
    t.sort()
    t = ''.join(t)
    return t

def all_anagrams(filename):
    """Finds all anagrams in a list of words.

    filename: string filename of the word list

    Returns: a map from each word to a list of its anagrams.
    """
    d = {}
    for line in open(filename):
        word = line.strip().lower()
        t = signature(word)

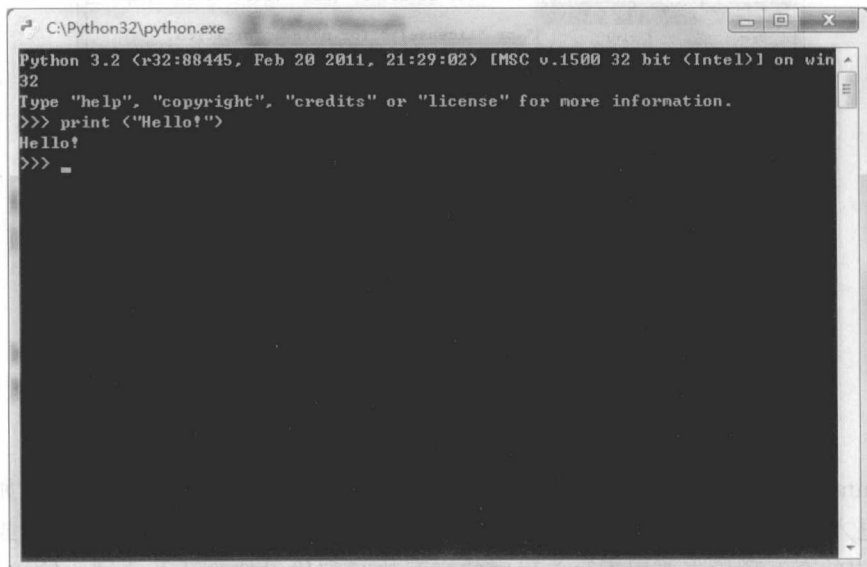
        if t not in d:
            d[t] = [word]
        else:
            d[t].append(word)
    return d

def print_anagram_sets(d):
    """Prints the anagram sets in d.

    d: map from words to list of their anagrams
    """
    for v in d.values():
        if len(v) > 1:
            print len(v), v

Ln: 1, Col: 0
```

图 1-6 新建窗口



```
C:\Python32\python.exe
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print <"Hello!">
Hello!
>>> =
```

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↑、↓可以上翻或下翻已经执行过的命令,以提高输入效率。

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

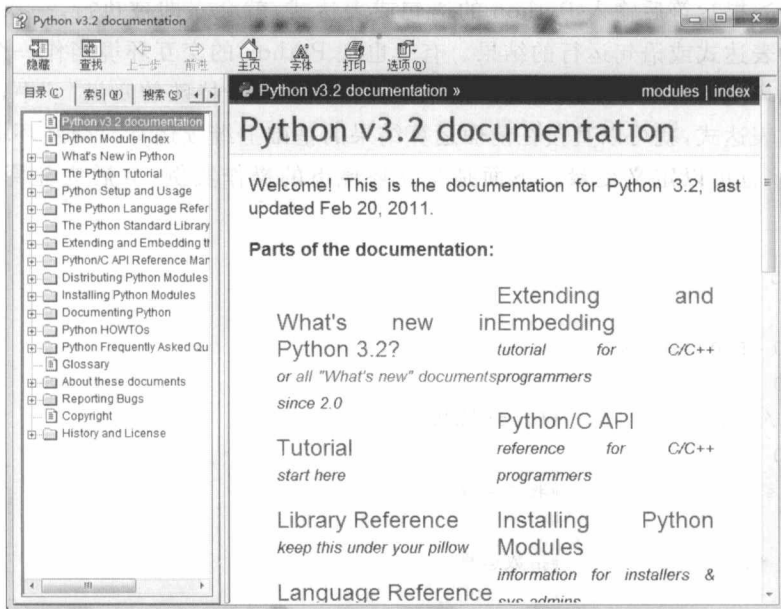


图 1-8 帮助窗口

在交互环境下同样可以使用 `dir` 和 `help` 命令来获得与 Python 的函数、对象属性、方法等有关的帮助信息。

1.2.2 Linux、UNIX 和 Macintosh

在 Linux 下的 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`,就出现 Python 的提示符 `>>>`。

1.3 执行 Python 程序的方法

运行 Python 脚本的方式有多种,主要方式为交互环境运行、命令行窗口运行,在 Windows 和 Linux 平台还有各自的特点。

1. 交互环境运行

安装 Python 后,有两种交互环境可运行脚本:一种是在命令行窗口(Command Line)输入 python,按 Enter 键;另一种是利用集成开发环境 IDLE,进入 IDLE 默认地打开交互环境窗口。

在 `>>>` 提示符后输入 Python 的语句或表达式,就会立即被执行,一次执行一条,正确时显示表达式或语句运行的结果。有人也把 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
>>>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
```

在 Python 中使用变量前必须先定义变量。上面的例子中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。