

普通高等教育“十三五”规划教材

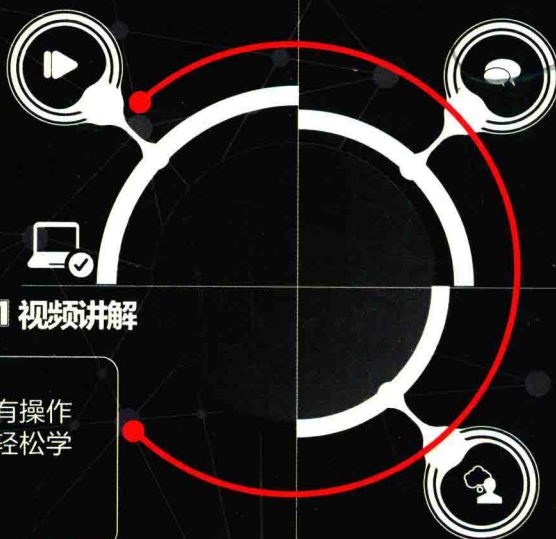
“互联网+”
新形态教材

{ 作者20多年应用软件开发、教材编写经验
经典的教程编写方式，案例驱动的学习情景 }

Python程序设计

视频讲解+实用案例+源代码+实训+考试系统

主 编 丁亚涛
副主编 王世好 胡继礼 阚峻岭



01 视频讲解
....

所有案例程序配有操作
视频讲解，扫码轻松学
会Python

案例经典、实用，融
汇所有知识点

02 实用案例
....

03 考试系统
....

集十多年开发经验，
集成命题+组卷+评
卷+考务软件系统



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

Python程序设计

主 编 丁亚涛

副主编 王世好 胡继礼 阚峻岭



中国水利水电出版社
www.waterpub.com.cn

• 北京 •

内 容 提 要

本书基于作者团队近几年的教学实践与研发经验，按照满足初学者对 Python 语言的需求进行编写。全书共 12 章，主要内容包括 Python 语言基础、结构化和面向对象程序设计、正则表达式、函数、文件、图形界面设计、网络程序设计、数据库、线程与进程、大数据技术等。

本书采用“案例驱动”的编写方式，以程序设计为中心，语法介绍精炼，内容叙述深入浅出、循序渐进，程序案例生动易懂，具有很好的启发性。每章均配备精心设计的习题。另外，本书所有程序配有视频讲解，教材配套提供题库及软件测试系统，可供平时练习、实验实训和课程测试之用。

本书既可以作为本专科院校 Python 语言程序设计课程的教材，也可以作为自学者的参考用书，同时可供各类考试人员复习参考。

本书提供电子教案和所有程序的源代码，读者可以从中国水利水电出版社网站免费下载，网址为：<http://www.waterpub.com.cn/softdown/>。

图书在版编目（C I P）数据

Python程序设计 / 丁亚涛主编. -- 北京：中国水利水电出版社，2018.9
普通高等教育“十三五”规划教材
ISBN 978-7-5170-7005-4

I. ①P… II. ①丁… 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(2018)第227297号

书 名	普通高等教育“十三五”规划教材 Python程序设计 Python CHENGXU SHEJI
作 者	主 编 丁亚涛 副主编 王世好 胡继礼 阚峻岭
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: sales@waterpub.com.cn 电话：（010）68367658（营销中心）
经 售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京智博尚书文化传媒有限公司
印 刷	三河市龙大印装有限公司
规 格	185mm×260mm 16开本 15印张 362千字
版 次	2018年9月第1版 2018年9月第1次印刷
印 数	0001—3000册
定 价	39.00元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换
版权所有·侵权必究

前言

preface

随着计算机技术的不断发展,高等教育的计算机教学一直在接受各种挑战,其中程序语言的选择也在不断变化之中。Python 语言作为一门新兴的语言受到用户广泛的关注和欢迎。很多高校陆续开设了 Python 程序设计课程,部分开设其他程序语言的课程也正在转向开设 Python 语言。

新的程序语言教学急需合适的教材。本书基于作者团队近几年的教学实践与研发经验,按照满足初学者对 Python 语言的需求编写而成。本书特色如下:

1. 立足基本语法

本书不求深度,但求实用,语法介绍立足基础、全面。书中很多案例都是经典实用的例子。“经典的就是最好的”,针对基础课程的教学来说更是如此。教材涉及到各种较为流行的技术,例如面向对象、大数据、爬虫、Web 开发、数据库技术等,虽然只是冰山一角,但指引和导向更为重要。

2. 重点自然突出

书中重要的知识点都作了重点介绍,并不回避难点,但强调“化难为易”,将难点、重点的掌握过程通过恰当的案例、注释和说明变得清晰而明了,从而减少对程序语言的畏难情绪,让读者感觉 Python 其实并不难学、Python 其实也很容易。对于选学内容,在标题前用*号标出。书中所有的代码都是精心设计的并都作了突出排版,所有案例都提供视频讲解,扫描对应的二维码即可在线观看。

3. 配附实训内容

本书第二部分是实训部分。作为一种新的尝试,本书将实训部分放在主教材之后,将理论学习和实训练习结合起来。

4. 配套练习题库及软件

参加编写的教师团队拥有十多年的考试系统开发和维护经验,在教材编写之前已经完成 Python 考试系统软件的开发,Python 的实验部分也在软件中实现,软件可以在练习之后自动生成电子的实验报告,这是本书的重要特色资源,也是目前大多数教材没有的。软件系统几乎包括命题、考务管理、成绩回收和分析统计等全部教学考试管理流程。

5. 资源导航

资源导航技术是解决资源即时共享的关键。本书在章节要点处标注了资源链接,方便

读者快速访问资源。另外，关于 Python 重要的网络资源也都经过优送后呈现给读者。

本书由丁亚涛任主编，王世好、胡继礼、阚峻岭任副主编。王世好编写 1、2 章，束建华编写 3、4 章，胡继礼编写 5、6 章，谭红春编写第 8 章，方芳编写第 11 章，丁亚涛编写 7、9、10、12 章和配套软件部分，谷宗运编写实训部分，欧阳婷编写习题和课件部分，阚峻岭负责文前、部分章节的审校。参加本书编写工作的还有杞宁、程一飞、韩静、朱薇、马春、金力、孙大勇、蔡莉、李芳芳、俞磊等。在全书的策划和出版过程中，得到多所高校从事教学工作的同仁的关心和帮助，他们对本书提出了很多宝贵的建议，在此表示感谢。中国水利水电出版社的领导和编辑对本书的编写和出版给予了大力支持和统筹策划，在此一并表示感谢。

本书所配电子教案及相关教学资源可以从中国水利水电出版社网站下载。使用本书的学校也可以与作者联系（E-mail：yataoo@126.com），索取更多相关教学资源，特别是考试系统平台。

由于编者水平有限，书中不足之处在所难免，敬请广大读者批评指正。

编 者
2018 年 7 月

目 录

Contents

理论部分

第 1 章 Python 概述	3	2.2.4 时间元组	32
1.1 程序设计与 Python 语言	3	2.3 字典	34
1.1.1 程序设计语言概述及程序设计的基本概念	3	2.3.1 字典对象的创建与删除	34
1.1.2 Python 语言的历史和发展	4	2.3.2 字典的使用	34
1.1.3 Python 解释器	5	2.3.3 字典类型的常用函数	35
1.1.4 Python 语言的特点	5	2.4 集合	36
1.2 安装与使用	5	2.4.1 集合对象的创建与删除	36
1.2.1 安装 Python 解释器	6	2.4.2 集合的使用	37
1.2.2 使用 Python	7	习题 2	39
1.2.3 使用 PyCharm	9	第 3 章 选择与循环	43
*1.2.4 同时安装多个 Python 版本	12	3.1 选择结构	43
1.3 基础知识	12	3.1.1 条件运算符	43
1.3.1 对象模型	12	3.1.2 单分支选择结构	44
1.3.2 基本数据类型	13	3.1.3 双分支选择结构	45
1.3.3 数据类型转换	14	3.1.4 多分支选择结构	46
1.3.4 常量、变量、运算符与表达式	15	3.1.5 选择结构的嵌套	48
1.3.5 内置函数	21	3.2 循环结构	51
1.3.6 基本输入输出	22	3.2.1 for 循环	51
*1.3.7 关于 JSON	23	3.2.2 while 循环	52
1.4 Python 代码的编写规范	23	3.2.3 else 语句	52
1.4.1 换行	23	3.2.4 break 和 continue 语句	53
1.4.2 缩进	24	3.3 综合案例	54
1.4.3 注释	24	习题 3	57
1.4.4 空行	25	第 4 章 字符串与正则表达式	61
习题 1	25	4.1 字符串	61
第 2 章 列表、元组、字典和集合	29	4.1.1 字符串编码格式	61
2.1 列表	29	4.1.2 字符串基本操作	63
2.1.1 列表的概念	29	4.2 正则表达式	68
2.1.2 列表的使用	29	4.2.1 基本语法	68
2.2 元组	31	4.2.2 正则表达式的应用	70
2.2.1 元组对象的创建与删除	32	4.3 综合案例	73
2.2.2 元组运算符	32	习题 4	74
2.2.3 元组与列表相互转换	32	第 5 章 函数	77
		5.1 函数的定义	77

IV ||| Python 程序设计

5.2 参数	78	第 8 章 图形界面设计	121
5.2.1 形参与实参	78	8.1 Tkinter	121
5.2.2 参数类型	78	8.2 控件	123
5.2.3 函数返回值	81	8.2.1 标签 (Label)	123
5.3 变量的作用域	82	8.2.2 框架 (Frame)	124
5.3.1 作用域	82	8.2.3 按钮 (Button)	125
5.3.2 global 和 nonlocal 关键字	82	8.2.4 输入框 (Entry)	126
5.4 lambda 表达式	83	8.2.5 单选按钮 (RadioButton) 和复选按钮 (CheckBox)	128
5.5 递归函数	84	8.2.6 消息 (Message) 和消息框 (MessageBox)	131
5.5.1 递归函数的形式	84	8.2.7 组合框 (ComboBox) 和列表框 (ListBox)	133
*5.5.2 汉诺塔游戏	85	8.2.8 滚动条 (Scrollbar)	134
*5.6 偏函数	86	8.2.9 绘图组件 (Canvas)	135
习题 5	87	*8.3 wxPython	135
第 6 章 面向对象程序设计	89	8.3.1 wxPython 基础知识	136
6.1 面向对象的概念	89	8.3.2 Frame 的概念与作用	136
6.2 类的定义与使用	89	习题 8	138
6.2.1 类的定义	89	第 9 章 网络编程	141
6.2.2 类的实例	90	9.1 计算机网络基础	141
6.2.3 类的属性	92	9.1.1 定义及分类	141
6.3 类的方法	93	9.1.2 网络体系结构及参考模型	141
6.3.1 类的常用内置方法	93	9.2 Socket 编程	143
6.3.2 方法的访问控制	94	9.3 嗅探器	146
6.3.3 方法的装饰器	95	9.4 抓取网页内容	147
6.4 继承	96	9.4.1 爬虫的基本流程	147
6.4.1 子类的 __init__() 方法	97	9.4.2 requests 库	147
6.4.2 子类添加新的属性和方法	97	9.5 网站开发	151
6.4.3 重写方法	99	9.6 综合案例	153
6.5 多态	101	习题 9	155
6.6 模块与包	102	第 10 章 大数据	157
6.6.1 模块简介	102	10.1 数据分析基础	157
6.6.2 模块的使用	102	10.1.1 pandas 简介	157
6.6.3 包	104	10.1.2 获取数据	159
习题 6	105	10.2 数据处理	161
第 7 章 文件	107	10.2.1 引入数据集	161
7.1 文件基本操作	107	10.2.2 查询数据	162
7.1.1 文件类型	107	10.3 大数据	163
7.1.2 文件的打开和关闭	108	10.3.1 定义	163
7.1.3 文件的读写	109	10.3.2 结构	164
7.1.4 文件的其他操作	110	10.3.3 意义	164
7.2 目录操作	111	10.3.4 大数据技术	165
7.2.1 目录操作	111	10.4 数据分析案例	165
7.2.2 OS 对象和 shutil 模块	112	10.4.1 随机数据分析	165
7.3 高级文件操作	113	10.4.2 实际数据分析	169
7.3.1 MD5	113	10.4.3 统计分析	170
7.3.2 文件比较	115		
7.3.3 Office 文档操作	115		
习题 7	119		

习题 10	173	习题 11	188
第 11 章 线程与进程	175	第 12 章 数据库编程	191
11.1 进程与线程的基本概念	175	12.1 概述	191
11.1.1 定义	175	12.2 MySQL	191
11.1.2 关系	175	12.2.1 PyMySQL	191
11.1.3 区别	175	12.2.2 数据库基本操作	193
11.2 Python 线程	176	12.2.3 数据库查询操作	194
11.2.1 threading 模块	176	12.2.4 数据库更新操作	195
11.2.2 线程锁 threading.RLock 和 threading. Lock	177	12.2.5 插入和删除操作	196
11.2.3 threading.Event	179	12.3 Access	197
11.2.4 threading.Condition	180	12.3.1 win32.client	197
11.2.5 queue 模块	181	12.3.2 pypysql 模块	199
11.3 Python 进程	183	*12.4 事务和错误处理	202
11.3.1 multiprocessing 模块	183	12.4.1 事务	202
11.3.2 进程池 (Pool)	185	12.4.2 错误处理	202
*11.4 Python 协程	187	习题 12	203

实训部分

实训 1 Python 语言基础	206	实训 7 文件操作	214
实训 2 数据类型、序列、字典和集合	207	实训 8 图形界面设计	216
实训 3 选择与循环	209	实训 9 网络程序设计	220
实训 4 字符串与正则表达式	211	实训 10 大数据	222
实训 5 函数设计与使用	212	实训 11 线程与进程	223
实训 6 面向对象程序设计	213	实训 12 数据库编程	225
附录 A Python 2 和 Python 3	226		
附录 B 常用的 Python 编辑环境	228		
附录 C Python 资源网站	229		
附录 D 关于配套资源的说明	231		
参考文献	232		

理论部分

第1章 Python 概述

第2章 列表、元组、字典和集合

第3章 选择与循环

第4章 字符串与正则表达式

第5章 函数

第6章 面向对象程序设计

第7章 文件

第8章 图形界面设计

第9章 网络编程

第10章 大数据

第11章 线程与进程

第12章 数据库编程

第 1 章 Python 概述

学习目标



扫一扫，看视频

- ◎ 理解计算机语言及程序设计的基本概念
- ◎ 了解 Python 语言的形成、发展和基本特点
- ◎ 掌握 Python 语言的基本数据类型、运算符和表达式
- ◎ 掌握基本输入输出的方法
- ◎ 了解 Python 代码的编写规范
- ◎ 熟悉 Python 语言的编程环境

1.1 程序设计与 Python 语言

1.1.1 程序设计语言概述及程序设计的基本概念

程序设计语言是用于书写计算机程序的语言。语言的基础是一组记号和一组规则。按照规则由记号构成的记号串的总体就是语言。在程序设计语言中，这些记号串就是程序。

程序设计语言包括三大类：机器语言、汇编语言和高级语言。

机器语言是由二进制 0、1 代码指令构成，不同的 CPU 具有不同的指令系统。机器语言程序难编写、难修改、难维护，需要用户直接对存储空间进行分配，编程效率极低。这种语言已经渐渐被淘汰了。

汇编语言指令是机器指令的符号化，与机器指令存在着直接的对应关系，所以汇编语言同样存在难学难用、容易出错、维护困难等缺点。但是汇编语言也有自己的优点：可直接访问系统接口，汇编程序翻译成的机器语言程序的效率高。从软件工程角度来看，只有在高级语言不能满足设计要求，或不具备支持某种特定功能的技术性能（如特殊的输入输出）时，汇编语言才被使用。

机器语言和汇编语言统称为低级语言。

高级语言是接近自然语言的一种计算机程序设计语言，可以更容易地描述计算问题并利用计算机解决问题。高级语言包括 C、C++、Java、Python、SQL 等。

高级语言按照计算机执行方式的不同可分成两类：静态语言和脚本语言。静态语言采用编译方式执行，如 C 语言、Java 语言等；脚本语言采用解释方式执行，如 JavaScript 语言、PHP 语言等。

编译是将高级语言编写的源代码转换成目标代码，执行编译的计算机程序称为编译器。计算机可以立即或稍后运行这个目标代码。如图 1-1 所示展示了编译过程。

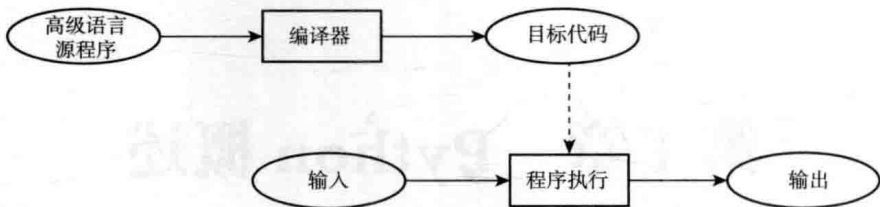


图 1-1 程序的编译和执行过程

解释是将高级语言编写的源程序逐条转换成目标代码的过程，执行解释的计算机程序称为解释器。如图 1-2 所示展示了解释过程。

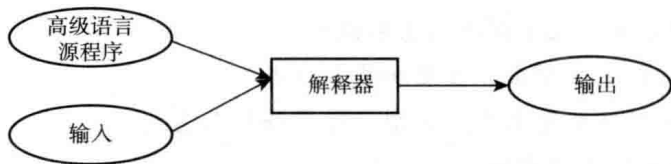


图 1-2 程序的解释和执行过程

编译和解释的区别在于，编译时一次性地将源程序转换成目标代码，一旦程序被编译，就不再需要编译程序或者源程序；解释则在每次程序运行时都需要解释器和源程序。

程序设计是用户根据具体的工作任务编写出能让计算机高效地完成该任务的程序的过程。程序设计一般包括以下几个部分：

- (1) 分析问题。对于接收的任务要进行认真分析，研究所给定的条件，分析最后应达到的目标，找出解决问题的规律，选择解题方法，完成实际问题。
- (2) 设计算法。即设计出解题的方法和具体步骤。
- (3) 编写程序。根据确定的算法，使用选择的计算机语言编写程序代码。
- (4) 运行程序，分析结果。运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，要对结果进行分析，看它是否合理。不合理要对程序进行调试，即通过上机发现和排除程序中的故障。
- (5) 编写程序文档。许多程序是提供给别人使用的，如同正式的产品应当提供产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书。程序文档的内容应包括：程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据，以及使用时的注意事项等。

1.1.2 Python 语言的历史和发展

Python 是由 Guido van Rossum 在 20 世纪 80 年代末至 90 年代初，在荷兰国家数学和计算机科学研究所设计出来的。Python 从一种教学语言 ABC 发展起来，并受到了 Modula-3（另一种相当优美且强大的语言，为小型团体设计）的影响。并且结合了 Unix shell 和 C 语言的习惯。

Python 语言是开源项目的优秀代表，其解释器的全部代码都是开源的，可以在 Python 语言的主网站（<https://www.python.org/>）上自由下载。

1991 年，第一个 Python 编译器诞生；2000 年，Python 2.0 发布；到 2010 年，Python 2.x 系列发布了最后一个版本 2.7。Python 2.7 将于 2020 年 1 月 1 日终止提供支持。用户如果想要在这个日期之后继续得到与 Python 2.7 有关的支持，则需要付费给商业供应商。

2008 年, Python 3.0 发布, 该版本在语法和解释器内部都做了很多改进, 这些修改导致 3.x 系列版本无法向下兼容 Python 2.0 系列的既有语法。因此, 所有基于 Python 2.0 系列版本编写的库函数都必须修改后才能被 Python 3.0 系列解释器运行。

1.1.3 Python 解释器

Python 规定了一个 Python 语法规则, 实现了 Python 语法的解释程序就成为了 Python 的解释器。Python 是一种高级通用的脚本编程语言, 虽然采用解释执行方式, 但它的解释器也保留了编译器的部分功能, 随着程序运行, 解释器也会生成一个完整的目标代码。

Python 的解释器主要有:

- (1) CPython (ClassicPython): 原始的 Python 实现, 是 C 语言实现的 Python, 是最常用的 Python 版本。
- (2) Jython: 原名 JPython, 是 Java 语言实现的 Python, 可以直接调用 Java 的各种函数库。
- (3) PyPy: 使用 Python 语言写的 Python 解释器。

另外还有 IronPython (面向 .NET 和 ECMA CLI 的 Python 实现)、ZhPy (支持使用繁/简体中文语句编写程序的 Python 语言) 等。

1.1.4 Python 语言的特点

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。具有以下典型的特点:

- (1) 易于学习。Python 有相对较少的关键字, 结构简单, 具有一个明确定义的语法, 学习起来更加简单。
- (2) 易于阅读。Python 代码的定义更清晰。
- (3) 易于维护。Python 的成功在于它的源代码是相当容易维护的。
- (4) 一个广泛的标准库。Python 的最大优势是有丰富的库, 可跨平台使用, 在 UNIX、Windows 和 Macintosh 环境的兼容性很好。
- (5) 互动模式。互动模式的支持, 可以从终端输入执行代码并获得结果的语言, 可以互动地测试和调试代码片断。
- (6) 与平台无关性。作为脚本语言, Python 程序可以在任何安装解释器的计算机环境中执行, Python 已经被移植到许多平台。
- (7) 可扩展。如果需要一段运行很快的关键代码, 或者是想要编写一些不愿开放的算法, 可以使用 C 或 C++ 完成那部分程序, 然后从 Python 程序中调用。
- (8) 数据库。Python 提供所有主要的商业数据库的接口。

1.2 安装与使用

Python 是一种跨平台的编程语言, 这意味着它能够运行在所有主要的操作系统中。在安装了 Python 的计算机上, 都能够运行 Python 程序。然而, 在不同的操作系统中, 安装 Python 的方法存在细微的差别。本书以 Python 3.6 版本为示例来讲述 Python 的安装。需要注意的是: 除了 Python 的安装以外, 对于不同的操作系统, 如 Windows、Unix、Linux, 本书的内容都是适用的。

1.2.1 安装 Python 解释器

Python 语言的解释器可以在 Python 语言的主网站下载，网址如下：

<https://www.python.org/downloads/>

1. Windows 下安装

选择 Downloads 菜单的 Windows 项，然后在列表中选择合适自己的版本并下载。双击所下载的程序安装 Python 解释器，然后出现如图 1-3 所示的引导过程。在该页面中，选中 Add Path 3.6 to PATH 复选框，这样能够更轻松地为系统配备。



图 1-3 安装程序的启动界面

读者也可以手工添加搜索路径。依次单击打开“系统属性->高级->环境变量”对话框，在用户变量或系统变量的 path 中添加 Python 安装路径，例如 C:\Python，如图 1-4 所示。

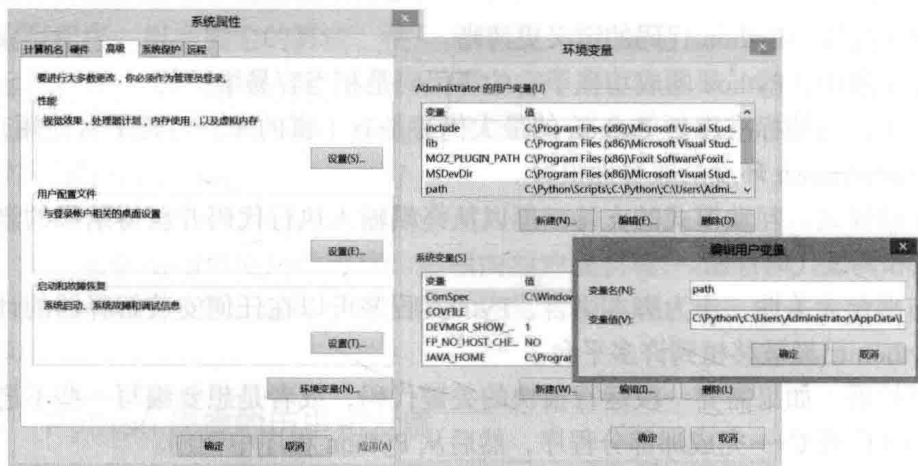


图 1-4 手工添加路径

Python 系统内置几个重要的环境变量，如表 1-1 所示。

表 1-1 Python 环境变量

变量名	描 述
PYTHONPATH	Python 搜索路径
PYTHONSTARTUP	Python 启动后，先寻找 PYTHONSTARTUP 环境变量，然后执行此变量指定的文件中的代码

续表

变量名	描 述
PYTHONCASEOK	加入 PYTHONCASEOK 的环境变量，就会使 Python 导入模块的时候不区分大小写
PYTHONHOME	另一种模块搜索路径。它通常内嵌于 PYTHONSTARTUP 或 PYTHONPATH 目录中，使得两个模块库更容易切换

2. Unix 与 Linux 下安装

几乎所有的 Linux 发行版中都预装了 Python 语言，如果需要安装的话，可参考下面的步骤：

打开 WEB 浏览器访问 <https://www.python.org/downloads/source/>，选择适用于 Unix/Linux 的源码压缩包。下载及解压压缩包，例如：

```
tar -xvzf Python-3.6.4.tgz
```

如果需要自定义一些选项可修改文档 Modules/Setup。

- 执行 ./configure 脚本
- 执行 make
- 执行 make install

执行以上操作后，Python 会安装在 /usr/local/bin 目录中，Python 库安装在如下位置：

```
/usr/local/lib/pythonXX
```

XX 为 Python 的版本号。

安装完成之后要简单做一下配置：将 Python 库路径添加到 /etc/ld.so.conf 配置中，然后执行 ldconfig 使其生效；或者添加到 \$LD_LIBRARY_PATH 中。

Python 安装包将在系统中安装一批与 Python 开发和运行有关的程序，其中最重要的两个是 Python 命令行和 Python 集成开发环境（IDLE）。

3. 安装 pip

运行 Python 时会经常需要安装第三方库文件，常用的方法是用 pip 来安装，如果没有 pip 的话，可以先安装 pip，安装的方法和步骤如下：

- (1) 下载 pip，地址为：<https://pypi.python.org/pypi/pip/downloads>。
- (2) 解压，例如解压到 C:\pip。
- (3) 在 Windows 命令行运行如下命令：

```
python c:\pip\setup.py install
```

1.2.2 使用 Python

对于初学者，Python 的使用在不同操作系统下的差别并不大。因此，本书的介绍以 Windows 操作系统下为主。

Python 程序的运行有两种方式：交互式和文件式。

1. 交互式启动和运行方法

第一种方法，启动 Windows 操作系统的命令行工具，单击“开始”按钮，在“开始”菜单的“搜索程序和文件”框中输入“cmd”命令，在出现的控制台中输入“Python”并回车，出现如图 1-5 所示的界面。

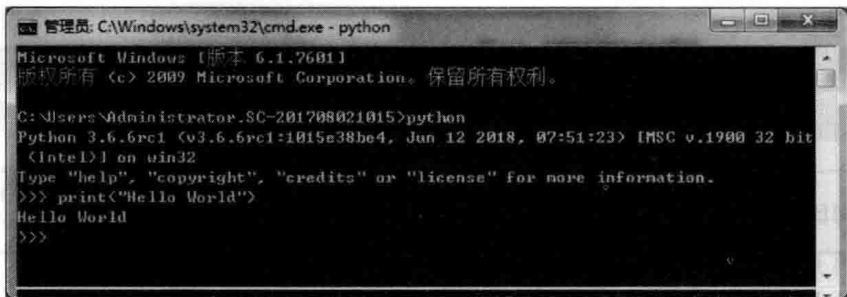


图 1-5 通过命令行启动交互式 Python 运行环境

在命令提示符 >>> 后面可以输入 Python 代码，例如，输入如下程序代码：

```
print("Hello World")
```

按回车键后显示如下结果：

```
Hello World
```

第二种方法，运行“开始”菜单中“所有程序”中 Python 的 IDLE。如图 1-6 所示的界面展示了 IDLE 环境中运行 print("Hello World") 程序的效果。

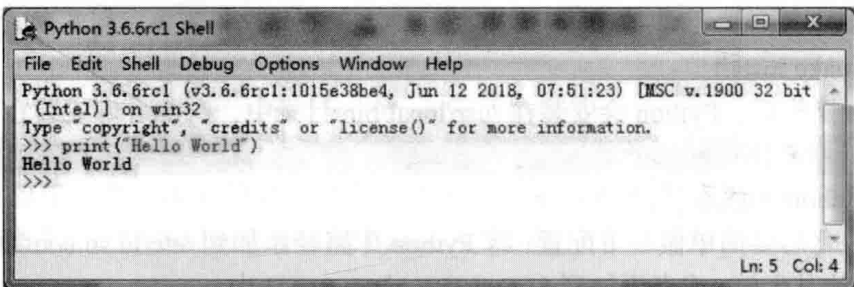


图 1-6 通过 IDLE 启动交互式 Python 运行环境

2. 文件式启动和运行方法

第一种方法，按照 Python 的语法格式，在任意编辑器中编写代码，并保存为扩展名为 .py 的文件。例如在 Windows 的记事本中输入程序代码 print("Hello World")，并以文件名 1-1.py 保存。然后打开 Windows 的命令 (cmd.exe)，进入 1-1.py 文件所在的文件夹，并运行该文件将获得输出结果，如图 1-7 所示。



图 1-7 通过命令行方式运行 Python 程序文件

第二种方法，打开 IDLE，在菜单中选择 File 菜单的 New File 选项，将出现一个具备 Python 语法高亮辅助的编辑器的新窗口，可以进行程序代码编辑，但是该窗口不是交互模式。例如，输入程序代码 print("Hello World")，并保存为 1-2.py 文件，如图 1-8 左图所示。

按 F5 键或选择 Run 菜单的 Run Module 选项运行该文件，结果如图 1-8 右图所示。

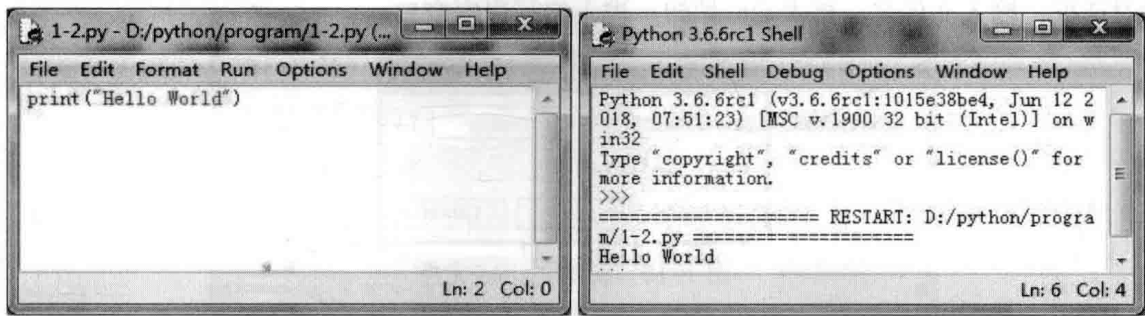


图 1-8 通过 IDLE 编写并运行 Python 程序文件

同样，在其他编辑环境中也可以通过打开文件来直接运行程序，例如下面要介绍的 PyCharm。

1.2.3 使用 PyCharm

1. PyCharm 简介

PyCharm 是由 JetBrains 打造的一款 Python IDE，带有一整套可以帮助用户在使用 Python 语言开发时提高效率的工具，比如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该 IDE 提供了一些高级功能，以用于支持 Django 框架下的专业 Web 开发。PyCharm 可以到其官网 <https://www.jetbrains.com/pycharm/> 下载和安装。

2. 在 PyCharm 中运行 Python 程序

(1) 创建工程。启动 PyCharm 后，选择 File 菜单的 New Project 命令，在弹出的窗口中选择 Pure Python，在 Location 处给出工程路径，然后单击 Create 按钮创建一个新的工程，如图 1-9 所示。

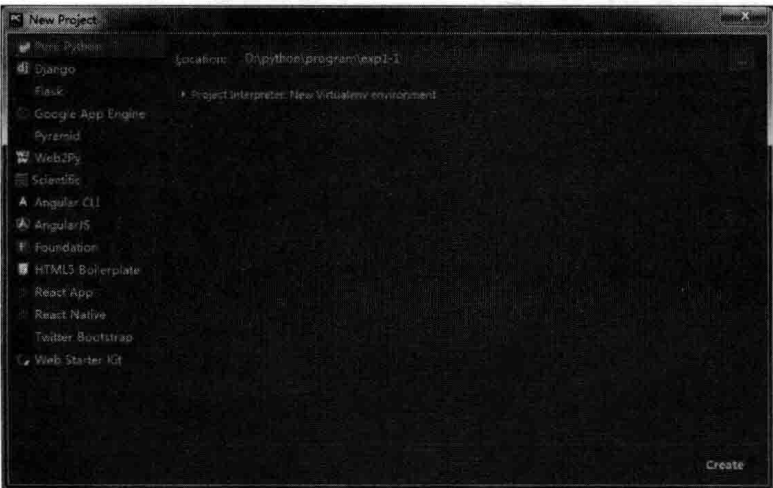


图 1-9 在 PyCharm 中创建新工程界面

另外，首次启动 PyCharm，在启动界面中选择 Create New Project 选项，也可以创建一个新的工程。