

# Python程序设计教程

蔡宝声 陈静◎主编



 北京理工大学出版社  
BEIJING INSTITUTE OF TECHNOLOGY PRESS

高等职业教育大数据技术与应用专业系列教材

# Python 程序设计教程

主 编 綦宝声 陈 静  
副主编 张 慧 彭福荣 许春秀 徐丽丽 卜令瑞

## 内 容 简 介

为了使广大读者既能够掌握 Python 语言的基础知识,又能够将 Python 语言应用于某个特定的领域(网络程序开发),本书介绍与 Python 相关的知识。为了便于读者学习,本书每个模块都提供了详尽的例子,结合实例讲解各个知识点。涉及的程序代码都给出了详细的注释,有助于读者轻松领会程序开发的精髓,快速提高开发技能。在学习完本书之后,相信读者能够很好地掌握 Python 语言,同时可以使用 Python 语言进行实际项目的开发。本书全部代码适用于 Python 3.7 及更高版本。

本书既可以作为研究生、本科生、专科生学习程序设计课程的教材,也可以作为 Python 程序设计爱好者的自学用书。

版权专有 侵权必究

---

### 图书在版编目(CIP)数据

Python 程序设计教程/綦宝声,陈静主编. —北京:北京理工大学出版社,2021.3  
(2021.4 重印)

ISBN 978-7-5682-9603-8

I. ①P… II. ①綦…②陈… 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(2021)第 041943 号

---

---

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 /

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 23.25

字 数 / 518 千字

版 次 / 2021 年 3 月第 1 版 2021 年 4 月第 2 次印刷

定 价 / 62.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 周瑞红

责任印制 / 施胜娟

---

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

# 前 言

作为最流行的脚本语言之一，Python 具有内置的高级数据结构和简单、有效的面向对象编程思想实现。同时，其语法简洁、清晰，类库丰富、强大，非常适合进行快速原型开发。另外，Python 可以运行在多种系统平台下，使得只需要编写一次代码，就可以在多个系统平台下保持同等的功能。

为了使广大读者既能够掌握 Python 语言的基础知识，又能够将 Python 语言应用于某个特定的领域（网络程序开发），本书将全面介绍和 Python 相关的内容。在学习完本书之后，读者应该能够很好地掌握 Python 语言，同时，可以使用 Python 语言进行实际项目的开发。

## 本书特点

### 1. 循序渐进，由浅入深

为了方便读者学习，本书首先让读者了解 Python 的历史和特点，再通过具体的例子逐渐把读者带入 Python 的世界，使他们掌握 Python 语言的基本知识要点及基础类库、常用库和工具的使用。

### 2. 技术全面，内容充实

本书在保证内容实用的前提下，详细介绍了 Python 语言的各个知识点。同时，本书所涉及的内容非常全面，无论从事什么行业的读者，都可以从本书中找到可以应用 Python 于自身所处行业的地方。

### 3. 代码完整，讲解详尽

书中的每个知识点都配有一段示例代码，代码的关键点也有注释说明。每段代码的后面都有详细的分析，同时给出了代码的运行结果。

### 4. 应用导向，能力训练

书中内容以程序设计应用为导向，突出使用 Python 解决实际问题的方法和能力训练。

### 5. 典型案例，综合练习

在各模块（不包括模块 1）中都精选和安排了与实际结合紧密的典型案例，让读者既可以通过基本案例学到 Python 的基础知识和使用方法，又可以通过典型案例对所学知识进行综合练习和应用，进一步提高编程能力。

## 内容组织与阅读建议

本书共有 15 个模块，主要包括 Python 编程基础（模块 1 ~ 9）和 Python 开发应用（模

块 10 ~ 15) 两部分内容, 全部代码适用于 Python 3.7 及更高版本。

模块 1 Python 语言概述。介绍 Python 语言与版本、开发环境安装与配置、编程规范、扩展库安装方法、标准库与扩展库中对象的导入与使用。

模块 2 Python 语言基础。包括基本输入/输出、基本数据类型、运算符等。

模块 3 Python 流程控制。包括选择结构的常见形式, 如单分支、双分支、多分支及嵌套的选择结构, for 循环与 while 循环, break 与 continue 语句。

模块 4 Python 序列。讲解 Python 中常用的数据结构, 包括列表、元组、字典、集合。

模块 5 函数。讲解函数的定义和调用、函数的参数、变量的作用域、lambda 表达式、生成器函数、递归函数、常用内置函数。

模块 6 面向对象程序设计。讲解创建类和对象、数据成员和成员方法、继承、特殊方法与运算符重载。

模块 7 字符串与正则表达式。包括字符串概述、字符串操作、正则表达式。

模块 8 文件操作。讲解文件读写操作、目录操作。

模块 9 异常处理。介绍 Python 中的异常、常用异常处理结构、断言语句。

模块 10 窗口界面设计。包括 tkinter 简介、tkinter 常用控件示例、布局管理、tkinter 事件。

模块 11 数据分析与处理。包括扩展库 pandas 介绍、pandas 数据类型、pandas 数据处理、可视化统计数据。

模块 12 网络编程。讲解 socket 编程、urllib 基本操作与爬虫案例、requests 基本操作与爬虫案例、scrapy 爬虫案例。

模块 13 数据可视化。讲解 Matplotlib 库、统计图形的绘制方法。

模块 14 Python 访问数据库。讲解 Python 访问 SQLite 数据库、Python 访问 Access 数据库、Python 访问 MySQL 数据库、Python 访问 MongoDB 数据库。

模块 15 进程和线程。包括创建进程、进程之间的通信、创建线程、线程同步。

本书既介绍了 Python 基础知识, 如 Python 编程基础、程序设计结构、函数、面向对象程序设计、常用标准库模块和常用第三方库等; 又介绍了 Python 中较为专业的内容, 如数据库访问、图形用户界面编程、多进程与多线程、网络程序设计等。

## 本书适用读者

本书既可以作为研究生、本科生、专科生学习程序设计课程的教材, 也可以作为 Python 程序设计爱好者自学用书。

## 致谢

本书由山东劳动职业技术学院信息工程系的綦宝声、陈静主编, 张慧、彭福荣、许春秀、徐丽丽、卜令瑞副主编。编者具有渊博的学识, 积累了丰富的教学经验, 其中, 模块 1、14 由綦宝声副教授编写, 模块 2、7、13 由张慧编写, 模块 3、8、11 由许春秀编写, 模块 4、9、12 由彭福荣编写, 模块 5、10、15 由徐丽丽编写, 模块 6 由陈静教授、卜令瑞编写。綦宝声副教授负责全书的总体结构设计和统稿工作。

本书采用了济南博赛网络技术有限公司、联想教育科技(北京)有限公司提供的部分

企业案例，大大丰富了读者学习的素材。另外，本书在编写过程中得到了山东女子学院数据科学与计算机学院孙洪峰院长、尹西杰教授的指导，在此表示感谢！

由于作者水平有限，书中疏漏之处在所难免，敬请同行及广大读者批评指正。

蔡宝声  
于山东济南  
2020年6月

# 目 录

<b>模块 1 Python 语言概述</b> .....	1
1.1 语言简介 .....	2
1.2 Python 版本选择 .....	2
1.3 Python 开发环境安装与配置 .....	3
1.3.1 IDLE .....	3
1.3.2 PyCharm .....	4
1.4 Python 编程规范 .....	5
1.5 扩展库安装方法 .....	6
1.6 标准库与扩展库中对象的导入与使用 .....	7
1.6.1 导入模块 .....	7
1.6.2 导入模块对象 .....	8
1.6.3 导入模块所有对象 .....	8
1.7 习题 .....	9
<b>模块 2 Python 语言基础</b> .....	11
2.1 基本输入/输出 .....	12
2.1.1 基本输出 .....	12
2.1.2 基本输入 .....	14
2.2 基本数据类型 .....	14
2.2.1 常量和变量 .....	14
2.2.2 数字类型 .....	15
2.2.3 字符串 .....	16
2.2.4 列表 .....	16
2.2.5 元组 .....	17
2.2.6 字典 .....	17
2.2.7 集合 .....	18
2.2.8 数据类型转换 .....	19
2.3 运算符 .....	20
2.3.1 算术运算符 .....	20
2.3.2 关系运算符 .....	21
2.3.3 逻辑运算符 .....	21



2.3.4	位运算符	22
2.3.5	成员运算符	23
2.3.6	身份运算符	23
2.3.7	赋值运算符	24
2.3.8	运算符的优先级	25
2.4	综合案例	25
2.5	习题	26
<b>模块 3</b>	<b>Python 流程控制</b>	<b>29</b>
3.1	选择结构	30
3.1.1	单分支选择结构	30
3.1.2	双分支选择结构	31
3.1.3	多分支选择结构	32
3.1.4	选择结构的嵌套	33
3.2	循环结构	35
3.2.1	for 循环和 while 循环	35
3.2.2	break 与 continue	38
3.3	综合案例	40
3.4	习题	42
<b>模块 4</b>	<b>Python 序列</b>	<b>45</b>
4.1	序列概述	46
4.1.1	序列索引	46
4.1.2	序列的内置函数	47
4.2	列表	47
4.2.1	列表创建和访问	48
4.2.2	列表添加元素	50
4.2.3	列表查询元素	52
4.2.4	列表删除元素	52
4.2.5	列表排序和反转	54
4.2.6	列表推导式	55
4.2.7	列表切片	56
4.2.8	列表对象支持的运算符	57
4.3	元组	58
4.3.1	元组创建和访问	59
4.3.2	元组更新和删除	60
4.3.3	生成器推导式	61
4.4	字典	62
4.4.1	字典创建和访问	62
4.4.2	字典添加与修改	66
4.4.3	字典删除元素	67

4.4.4	字典应用案例 .....	67
4.5	集合 .....	68
4.5.1	集合创建和访问 .....	68
4.5.2	集合操作与运算 .....	70
4.5.3	集合对象支持的运算符 .....	71
4.5.4	集合应用案例 .....	71
4.6	习题 .....	72
<b>模块 5</b>	<b>函数</b> .....	<b>73</b>
5.1	函数的定义和调用 .....	74
5.1.1	定义函数 .....	74
5.1.2	调用函数 .....	76
5.2	函数的参数 .....	76
5.2.1	位置参数 .....	77
5.2.2	默认值参数 .....	77
5.2.3	关键字参数 .....	79
5.2.4	可变长度参数 .....	80
5.2.5	解包参数列表 .....	83
5.2.6	特殊参数 .....	86
5.2.7	传值与传址参数 .....	86
5.3	变量的作用域 .....	88
5.4	lambda 表达式 .....	90
5.5	生成器函数 .....	91
5.6	递归函数 .....	92
5.7	常用内置函数 .....	95
5.7.1	数学计算相关函数 .....	96
5.7.2	迭代相关函数 .....	98
5.7.3	系统函数 .....	103
5.7.4	常用高阶函数 .....	108
5.8	综合案例 .....	110
5.9	习题 .....	112
<b>模块 6</b>	<b>面向对象程序设计</b> .....	<b>115</b>
6.1	创建类和对象 .....	116
6.2	数据成员和成员方法 .....	118
6.2.1	私有成员与共有成员 .....	118
6.2.2	数据成员 .....	119
6.2.3	方法 .....	120
6.2.4	属性 .....	122
6.3	继承 .....	124
6.4	特殊方法与运算符重载 .....	127



6.5 综合案例 .....	129
6.5.1 自定义数组 .....	129
6.5.2 自定义矩阵 .....	136
6.5.3 自定义队列 .....	138
6.6 习题 .....	140
<b>模块7 字符串与正则表达式 .....</b>	<b>141</b>
7.1 字符串概述 .....	142
7.1.1 字符串编码 .....	142
7.1.2 转义字符 .....	142
7.2 字符串操作 .....	144
7.2.1 字符串格式化 .....	144
7.2.2 字符串运算 .....	145
7.2.3 字符串常用方法 .....	148
7.3 正则表达式 .....	157
7.3.1 正则表达式语法 .....	157
7.3.2 正则表达式模块 re .....	158
7.3.3 正则表达式对象 match .....	159
7.4 综合案例 .....	161
7.5 习题 .....	162
<b>模块8 文件操作 .....</b>	<b>163</b>
8.1 文件读写操作 .....	164
8.1.1 打开文件 .....	164
8.1.2 关闭文件 .....	165
8.1.3 读文本文件 .....	166
8.1.4 写文本文件 .....	167
8.1.5 读写二进制文件 .....	168
8.1.6 定位读写位置 .....	174
8.1.7 复制文件 .....	174
8.1.8 移动文件 .....	176
8.1.9 重命名文件 .....	176
8.1.10 删除文件 .....	176
8.2 目录操作 .....	177
8.2.1 创建目录 .....	178
8.2.2 获取目录 .....	179
8.2.3 遍历目录 .....	179
8.2.4 删除目录 .....	181
8.3 综合案例 .....	182
8.4 习题 .....	186
<b>模块9 异常处理 .....</b>	<b>187</b>

9.1 Python 中的异常 .....	188
9.1.1 异常概念 .....	188
9.1.2 异常类型 .....	189
9.2 常用异常处理结构 .....	190
9.2.1 try...except...结构 .....	190
9.2.2 try...except...else...结构 .....	192
9.2.3 try...except...finally...结构 .....	194
9.2.4 raise 抛出异常 .....	195
9.3 断言语句 .....	195
9.4 习题 .....	196
<b>模块 10 窗口界面设计 .....</b>	<b>197</b>
10.1 tkinter 简介 .....	198
10.1.1 安装和使用 tkinter .....	198
10.1.2 创建第一个窗口界面 .....	199
10.1.3 tkinter 常用控件 .....	200
10.1.4 控件标准属性 .....	201
10.2 tkinter 常用控件示例 .....	202
10.2.1 按钮类控件 .....	202
10.2.2 文本输入类控件 .....	210
10.2.3 菜单及滚动条控件 .....	213
10.2.4 框架类控件 .....	216
10.3 布局管理 .....	221
10.3.1 pack 方式 .....	222
10.3.2 place 方式 .....	222
10.3.3 grid 方式 .....	224
10.4 tkinter 事件 .....	225
10.5 综合案例 .....	228
10.6 习题 .....	231
<b>模块 11 数据分析与处理 .....</b>	<b>233</b>
11.1 扩展库 pandas .....	234
11.1.1 pandas 的数据结构 .....	234
11.1.2 pandas 的安装 .....	234
11.2 pandas 数据类型 .....	235
11.2.1 Series .....	235
11.2.2 DataFrame 数据类型 .....	243
11.2.3 pandas 数据类型的基本运算 .....	251
11.3 pandas 数据类型的数据处理 .....	255
11.3.1 数据导入 .....	255
11.3.2 数据导出 .....	256



11.3.3	统计分析	256
11.3.4	数据处理	258
11.4	可视化统计数据	260
11.4.1	绘图库 Matplotlib	260
11.4.2	pyplot 基础图形绘制	270
11.5	综合案例	272
11.6	习题	276
<b>模块 12</b>	<b>网络编程</b>	<b>277</b>
12.1	socket 编程	278
12.1.1	UDP 协议编程	279
12.1.2	TCP 协议编程	282
12.2	urllib 基本操作与爬虫案例	285
12.2.1	urllib 基本操作	285
12.2.2	urllib 爬虫案例	288
12.3	requests 基本操作与爬虫案例	289
12.3.1	requests 基本操作	290
12.3.2	requests 爬虫案例	293
12.4	scrapy 爬虫案例	294
12.5	习题	297
<b>模块 13</b>	<b>数据可视化</b>	<b>299</b>
13.1	Matplotlib 库	300
13.1.1	Matplotlib 概述	300
13.1.2	Matplotlib 安装	300
13.2	绘制统计图形	301
13.2.1	折线图	301
13.2.2	散点图	302
13.2.3	直方图	304
13.2.4	条形图	305
13.2.5	箱形图	307
13.2.6	饼图	309
13.3	综合案例	311
13.4	习题	316
<b>模块 14</b>	<b>Python 访问数据库</b>	<b>317</b>
14.1	Python 访问 SQLite 数据库	318
14.1.1	Connection 对象	319
14.1.2	Cursor 对象	320
14.1.3	Row 对象	324
14.2	Python 访问 Access 数据库	325
14.3	Python 访问 MySQL 数据库	327

---

14.4	Python 访问 MongoDB 数据库 .....	333
14.5	习题 .....	342
<b>模块 15</b>	<b>进程和线程 .....</b>	<b>343</b>
15.1	创建进程 .....	344
15.1.1	Multiprocessing 模块 .....	344
15.1.2	Process 类 .....	344
15.1.3	Pool 进程池 .....	345
15.2	进程之间的通信 .....	347
15.2.1	Queue .....	347
15.2.2	Pipe .....	349
15.3	创建线程 .....	351
15.3.1	threading 模块 .....	351
15.3.2	Thread 类 .....	351
15.4	线程同步 .....	353
15.4.1	多线程抢夺变量 .....	353
15.4.2	线程锁 .....	354
15.4.3	线程优先队列 .....	355
15.5	习题 .....	356
	参考文献 .....	357



# 模块 1

---

## Python 语言概述

Python 语言以快速解决问题而著称，其特点在于提供了丰富的内置对象、数据结构和标准库对象，而庞大的扩展库更是极大增强了 Python 的功能，大幅度拓展了 Python 的用武之地，其应用几乎已经渗透到了所有学科和领域。本模块将介绍 Python 语言的特点、版本、编码规范、扩展库的安装、标准库对象与扩展库对象的导入和使用。

### 本模块学习目标

- 了解 Python 语言版本
- 熟悉 Python 开发环境
- 了解 Python 编码规范
- 掌握扩展库安装方式
- 掌握标准库对象与扩展库对象的导入和使用

## 1.1 语言简介

Python 语言是由 Guido van Rossum（吉多·范罗苏姆）在 1989 年开发的，Python 语言的名字来自一个著名的电视剧“Monty Python’s Flying Circus”，创作者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。

Python 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，从而提高加载速度并对源代码进行一定程度的保密，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各种平台上的可执行文件；Python 支持命令式编程和函数式编程两种方式，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。最重要的是，其拥有大量的几乎支持所有领域应用开发的成熟扩展库。

Python 语言拥有强大的“胶水”功能，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。Python 自诞生以来，在 30 多年的时间里，已经渗透到云计算、大数据分析、Web 前端开发、统计分析、移动终端开发、科学计算、系统运维、人工智能、机器学习、密码学、计算机辅助教学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。

## 1.2 Python 版本选择

Python 的官方网站是 <https://www.python.org/>，其上同时发行和维护着 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。目前常用版本分别是 Python 2.7.17、Python 3.7.6 和 Python 3.8.2。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，适用于 Python 2.x 和 Python 3.x 的扩展库之间更是差别巨大，这也是旧系统进行版本迁移时最大的障碍。

Python 3.x 的设计理念更加合理、高效和人性化，代码开发和运行效率更高。从 2015 年年底开始，Python 3.x 就已经呈现出全面普及和应用的趋势，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。另外，Python 官方早在 2016 年就已经宣布，最迟到 2020 年将会全面放弃 Python 2.x 的维护和更新，但目前看仍然在保持更新。

在选择 Python 时候，一定要先考虑清楚自己的学习目的，例如，打算做哪方面的开发、需要用到哪些扩展库，以及扩展库支持的最高 Python 版本等，明确这些问题后再做出适合自己的选择。如果刚刚开始接触 Python，那么一定要毫不犹豫地选择 Python 3.x 版本。

## 1.3 Python 开发环境安装与配置

除了 Python 官方安装包自带的 IDLE，还有 Anaconda3、PyCharm、Eclipse、zwPython 等大量开发环境。相对来说，IDLE 稍微简陋一些，但也提供了语法高亮（使用不同的颜色显示不同的语法元素，例如，使用绿色显示字符串、橙色显示 Python 关键字、紫色显示内置函数）、交互式运行、程序编写和运行及简单的程序调试功能。其他 Python 开发环境则是对 Python 解释器主程序进行了不同的封装和集成，使得代码的编写和项目管理更加方便一些。本节对 IDLE 和 PyCharm 这两个开发环境进行简单介绍，但书中所有代码也同样可以在 Anaconda3 等其他开发环境中运行。

按照惯例，本书中所有在交互模式运行和演示的代码都以 IDLE 交互环境的提示符“>>>”开头，在运行这样的代码时，并不需要输入提示符“>>>”。而书中所有不带提示符“>>>”的代码都表示需要写入一个程序文件并保存和运行。

### 1.3.1 IDLE

IDLE 应该算是最原始的 Python 开发环境之一，没有集成任何扩展库，也不具备强大的项目管理功能。但也正是因为这一点，使得开发过程中的一切尽在自己掌握中，深得资深 Python 爱好者的喜爱，成为 Python 内功修炼的重要途径。

在 Python 官方网站 <https://www.python.org/> 下载最新的 Python 3.7.x 安装包或 Python 3.8.x（根据自己计算机操作系统选择 32 位或 64 位）并安装（建议安装路径为 D:\Python37 或 D:\Python38）后，在“开始”菜单中可以打开 IDLE，如图 1-1 所示，然后看到的界面就是交互式开发环境，如图 1-2 所示。

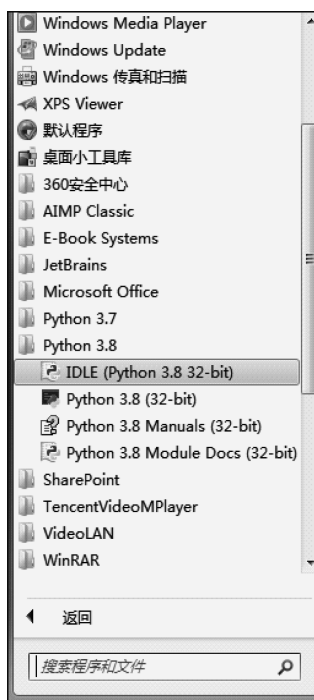


图 1-1 “开始”菜单

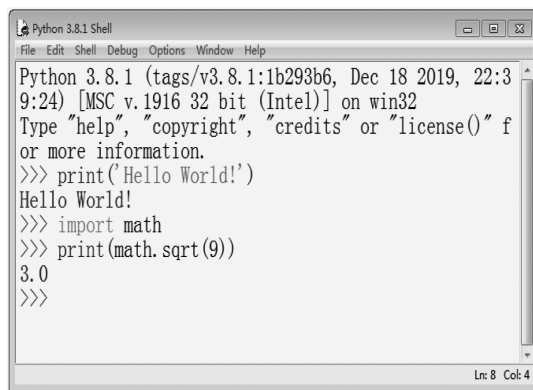


图 1-2 IDLE 交互式开发界面

在交互式开发环境中，每次只能执行一条语句，当提示符“>>>”再次出现时方可输入下一条语句。普通语句可以直接按 Enter 键运行并立刻输出结果，而选择结构、循环结构、函数定义、类定义、with 块等属于一条复合语句，需要按两次 Enter 键才能执行。如果要重复使用上面的命令，可以按 Alt + P 组合键。

如果要执行大段代码，也为了方便反复修改，可以在 IDLE 中选择“File”→“New File”命令来创建一个程序文件，将其保存为扩展名为 .py 或 .pyw 的文件，然后按 F5 键或选择“Run”→“Run Module”命令运行程序，结果会显示到交互式窗口中，如图 1-3 所示。

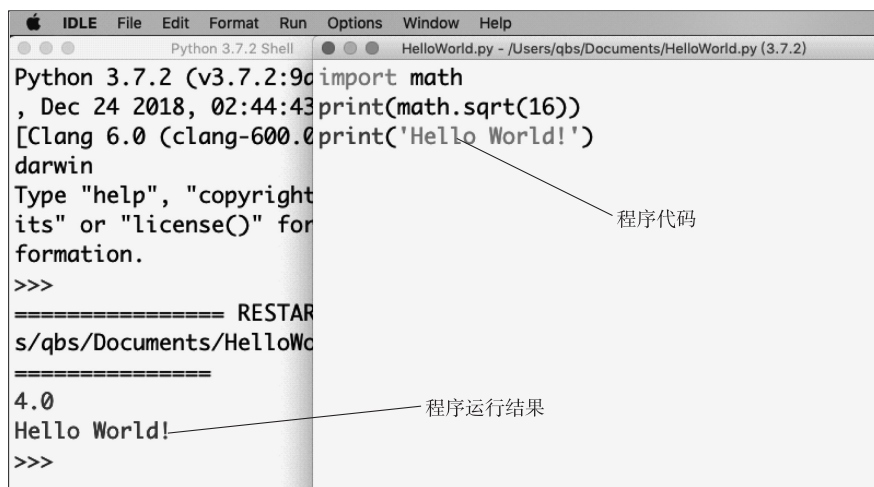


图 1-3 使用 IDLE 编写和运行 Python 程序

### 1.3.2 PyCharm

PyCharm 是一种 Python IDE，可以到官网 <https://www.jetbrains.com/pycharm/download/> 进行下载，其带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，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、版本控制等。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。同时支持 Google App Engine，更重要的是，PyCharm 支持 IronPython。IronPython 是优雅的 Python 编程语言和强大的 .NET 平台的有机结合。这些功能在先进代码分析程序的支持下，使 PyCharm 成为 Python 专业开发人员和刚起步人员使用的有力工具。

在“开始”菜单中可以打开 PyCharm，然后看到的界面就是 IDE 开发环境，如图 1-4 所示。整个界面包括菜单栏、工具栏、文件导航区、程序编辑区、控制台输出区五个部分，菜单栏包括 PyCharm 的所有菜单控制命令，工具栏用来对程序的运行和调试进行控制，文件导航区对项目包含的文件、目录进行管理，包括创建、删除、重命名等，程序编辑区用来编写程序代码，控制台输出区用来显示程序运行的结果。