

大数据创新人才培养系列

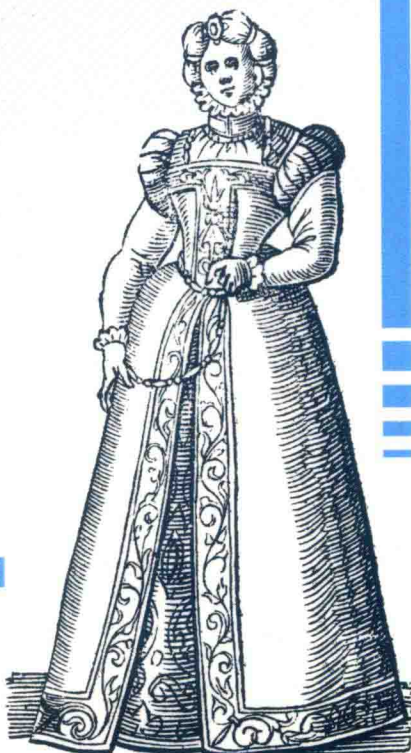
COMPUTER

# Python 程序设计 教程

Python Programming Course

陈沛强 主编

- 零基础入门，快速掌握 Python 3 的基本技术
- 采用案例驱动的编写模式，使读者轻松上手
- 丰富的配套资源，便于读者全方位学习教程



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

大数据创新人才培养系列

COMPUTER

# Python 程序设计 教程

Python Programming Course

■ 陈沛强 主编



人民邮电出版社

北京

## 图书在版编目 (CIP) 数据

Python程序设计教程 / 陈沛强主编. -- 北京: 人民邮电出版社, 2019.1

(大数据创新人才培养系列)

ISBN 978-7-115-49462-7

I. ①P… II. ①陈… III. ①软件工具—程序设计—教材 IV. ①TP311.561

中国版本图书馆CIP数据核字(2018)第224973号

## 内 容 提 要

本书系统地讲解了 Python 语言程序设计的相关知识。全书共有 12 章, 内容包括: Python 语言简介, 数据类型、运算符及表达式, 程序流程控制, 序列, 映射和集合类型, 函数, 模块和包, 文件, 面向对象编程, Python 数据库编程, tkinter GUI 编程, Django 框架 Web 编程。为了让读者能够及时地检查自己的学习效果, 把握自己的学习进度, 每章后面都附有习题和上机练习。

本书既可以作为高校 Python 语言程序设计课程的教材, 也可以作为对 Python 感兴趣的技术人员的参考资料。

---

◆ 主 编 陈沛强

责任编辑 李 召

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

山东百润本色印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 14.75

2019 年 1 月第 1 版

字数: 385 千字

2019 年 1 月山东第 1 次印刷

---

定价: 49.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

Python 已经成为受欢迎的程序设计语言之一。尤其自 2004 年以后, Python 的使用率更是呈线性增长。随着大数据时代的到来, 由于 Python 语言的简洁性、易读以及可扩展性, 用 Python 做科学计算和大数据分析的研究和商业机构日益增多, 在国外更是早已普及, 很多国外知名大学率先采用 Python 教授程序设计课程。国内近几年已经认识到了 Python 语言的重要性, 随着各行业使用 Python 语言的增多, 各高校也开始逐渐开设了该门课程。但目前适合作为国内高校 Python 语言程序设计课程的教材还比较少, 且已有的教材存在如下一些问题。

(1) 部分教材偏重理论, 缺乏实用性。

(2) 有些教材过于繁难或过于简单, 对于大学生的帮助不是很大。

(3) 现有教材大多面向 Python 2.x, 而面向 Python 3.x 的也存在内容不全面的问题。

本书的编写正是基于解决以上的诸多问题, 提供一本真正适合于我国高校教学特点的 Python 程序设计指导。本书以最新的 Python 3.x 为主线, 同时考虑了 Python 2.x 的特点, 不但给与适当的理论阐述, 更加注重 Python 语言不同于 Java 等其他主流语言特点的论述, 更加注重实用性, 案例丰富, 让读者更易上手。本书内容全面, 涉及面向过程、面向对象、数据库编程、窗口编程和 Web 编程的所有知识和技术。

本书完全采用案例驱动的编写模式, 每章都提供了上机练习, 这些上机练习也可作为实验课教学内容。本书采用一个项目贯穿始终, 可让读者体会到一个项目采用不同技术编程实现的异同, 既适合初学者夯实基础, 又能帮助 Python 程序员提升技能, 还可以作为 Python 程序员手边的工具书。

本书的参考学时为 48~64 学时, 建议采用理论实践一体化教学模式, 各章的参考学时见下面的学时分配表。

学时分配表

课 程 内 容	学 时
Python 语言简介	2
数据类型、运算符及表达式	2
程序流程控制	2
序列	4~6
映射和集合类型	4~6
函数	4~6
模块和包	4
文件	4~6
面向对象编程	6~8
Python 数据库编程	4~6

续表

课 程 内 容	学 时
tkinter GUI 编程	6~8
Django 框架 Web 编程	6~8
课时总计	48~64

由于编者水平和经验有限，书中难免有欠妥和疏漏之处，恳请读者批评指正。

编 者  
2018 年 7 月

# 目 录

第 1 章 Python 语言简介	1	第 3 章 程序流程控制	24
1.1 什么是 Python 语言	1	3.1 控制结构	24
1.2 Python 语言的发展历史	1	3.2 选择结构	24
1.3 Python 语言的特点	2	3.2.1 if 选择结构	25
1.4 Python 语言的应用	3	3.2.2 if/else 选择结构	26
1.5 Python 的安装	4	3.2.3 if/elif/else 选择结构	27
1.6 第一个 Python 程序	5	3.3 循环结构	29
小结	7	3.3.1 while 循环结构	29
习题	7	3.3.2 for 循环结构	31
第 2 章 数据类型、运算符及表达式	8	小结	33
2.1 Python 程序基本结构	8	习题	33
2.2 基本输入和输出	9	第 4 章 序列	35
2.2.1 基本输入	10	4.1 字符串	35
2.2.2 基本输出	10	4.1.1 字符串的表示和创建	35
2.3 数据类型	11	4.1.2 字符串基本操作	37
2.3.1 变量	11	4.1.3 字符串方法	39
2.3.2 数据类型：数字常量和数字对象	12	4.2 列表	42
2.4 运算符	15	4.2.1 列表基本操作	42
2.4.1 算术运算符和算术表达式	15	4.2.2 列表的访问、排序和反转	44
2.4.2 关系运算符和关系表达式	16	4.3 元组	46
2.4.3 逻辑运算符和逻辑表达式	18	4.3.1 元组的创建	46
2.4.4 位运算符	19	4.3.2 元组的访问	47
2.4.5 赋值运算符和赋值表达式	20	4.3.3 元组的其他基本操作和方法	47
2.4.6 其他运算符	21	小结	48
2.4.7 运算符的优先级	22	习题	49
小结	23	第 5 章 映射和集合类型	50
习题	23	5.1 映射类型——字典	50
		5.1.1 创建字典	50

5.1.2 访问字典	52	7.2.1 导入模块	79
5.1.3 更新字典	53	7.2.2 导入与执行语句	80
5.1.4 字典常用操作和方法	55	7.2.3 import 及 from 的使用	81
5.2 集合类型	57	7.2.4 重新载入模块	83
5.2.1 创建集合	57	7.2.5 嵌套导入模块	83
5.2.2 访问集合	58	7.2.6 模块对象属性和命令行 参数	84
5.2.3 更新集合	58	7.2.7 模块搜索路径	85
5.2.4 集合操作	61	7.3 包	87
小结	61	7.3.1 包的基本结构	88
习题	62	7.3.2 包的导入	88
<b>第 6 章 函数</b>	<b>64</b>	小结	90
6.1 函数的定义	64	习题	90
6.2 函数的调用	65	<b>第 8 章 文件</b>	<b>92</b>
6.3 函数参数	65	8.1 文件的打开与关闭	92
6.3.1 参数的多态性	65	8.2 文件的读写	94
6.3.2 参数的传递	66	8.2.1 文件的读取	94
6.4 函数的嵌套定义和 lambda 函数 定义	68	8.2.2 文件的写入	97
6.5 函数的递归调用和函数序列 调用	69	8.2.3 文件存储 Python 对象	98
6.6 函数变量的作用域	70	8.3 文件的备份和删除	99
6.6.1 变量的作用域	70	8.3.1 文件的备份	99
6.6.2 全局变量和局部变量	70	8.3.2 文件的删除	100
6.6.3 global 语句	72	8.4 文件夹的创建和删除	101
6.6.4 nonlocal 语句	72	8.4.1 文件夹的创建	101
6.6.5 实例：函数库	73	8.4.2 文件夹的删除	102
小结	74	小结	103
习题	75	习题	103
<b>第 7 章 模块和包</b>	<b>76</b>	<b>第 9 章 面向对象编程</b>	<b>105</b>
7.1 命名空间	76	9.1 理解 Python 的面向对象	105
7.1.1 命名空间的分类	76	9.1.1 Python 的类	105
7.1.2 命名空间的规则	77	9.1.2 Python 中的对象	106
7.1.3 命名空间示例	77	9.2 定义和使用类	106
7.2 模块	78	9.2.1 定义类	106
		9.2.2 使用类	107

9.2.3 实例	108	10.2.2 连接 MySQL 服务器	152
9.3 类、对象的属性和方法	109	10.2.3 MySQL 数据库操作	154
9.3.1 属性	110	10.2.4 MySQL 表操作	155
9.3.2 方法	113	10.2.5 MySQL 查询参数	156
9.4 类的组合	117	10.3 实例：加入数据库的用户注册 信息系统	157
9.5 继承与派生	120	小结	162
9.5.1 派生类的定义	120	习题	162
9.5.2 super 和方法重载	121	<b>第 11 章 tkinter GUI 编程</b>	163
9.5.3 多重继承	123	11.1 tkinter 编程基础	163
9.6 运算符重载和模块中的类	125	11.1.1 第一个 tkinter GUI 程序	163
9.6.1 运算符重载	125	11.1.2 组件打包	165
9.6.2 模块中的类	126	11.1.3 添加按钮和事件处理 函数	167
9.7 异常处理	127	11.1.4 使用布局	169
9.7.1 Python 异常处理机制	127	11.1.5 使用框架	171
9.7.2 主动引发异常	132	11.2 tkinter 组件	172
9.7.3 自定义异常类	133	11.2.1 组件通用属性设置	172
9.8 实例：用户注册信息	134	11.2.2 输入组件 Entry	175
9.8.1 功能预览	134	11.2.3 列表框组件 Listbox	178
9.8.2 功能实现	137	11.2.4 复选框组件 Checkbutton	179
小结	142	11.2.5 标签框架 LabelFrame	180
习题	142	11.2.6 文本框组件 Text	182
<b>第 10 章 Python 数据库编程</b>	144	11.2.7 顶层窗口组件 Toplevel	184
10.1 访问 SQLite 数据库	144	11.2.8 菜单组件 Menu	185
10.1.1 了解 Python 的 SQL 接口	144	11.2.9 对话框	187
10.1.2 连接和创建 SQLite 数据库	145	11.3 实例：基于 GUI 的用户注册 信息系统	190
10.1.3 创建表	145	11.3.1 系统功能预览	190
10.1.4 添加记录	146	11.3.2 显示全部已注册用户	191
10.1.5 执行查询	146	11.3.3 查找/修改/删除用户信息	191
10.1.6 使用 Row 对象	148	11.3.4 添加新用户	192
10.1.7 修改记录	148	11.3.5 系统功能实现	192
10.1.8 删除记录	149	小结	198
10.1.9 实例：导入文件中的 数据	150	习题	198
10.2 访问 MySQL 数据库	151		
10.2.1 访问 MySQL 数据库实例	151		



## 第 12 章 Django 框架 Web 编程

12.1 Django 框架与 MTV 模式	199	12.3.3 设计数据模型	206
12.1.1 Django 框架简介	199	12.3.4 数据迁移	207
12.1.2 Django 的 MTV 模式	200	12.4 Template 模板	208
12.1.3 Django 安装	200	12.4.1 什么是模板	208
12.2 创建 Django 项目	201	12.4.2 模板的继承	209
12.2.1 创建开发项目	201	12.4.3 静态文件服务	211
12.2.2 运行开发服务器	202	12.5 实例：学生信息管理	213
12.3 Django 项目的高级配置	203	12.5.1 查询学生	213
12.3.1 创建项目应用	203	12.5.2 添加学生	215
12.3.2 配置文件	205	12.5.3 修改学生	220
		12.5.4 删除学生	224
		小结	227
		习题	227

# 第 1 章

## Python 语言简介

### 本章要点

- 了解 Python 语言的发展历史。
- 掌握 Python 语言的特点。
- 了解 Python 语言的应用。
- 掌握 Python 语言中的安装。
- 学习第一个 Python 程序。

本章向读者介绍有关 Python 的背景知识，包括什么是 Python 以及它的发展历史，然后介绍 Python 语言的特色、应用领域，最后介绍 Python 的安装以及第一个 Python 程序。

## 1.1 什么是 Python 语言

Python 是一种简单易学、面向对象、解释型的计算机程序设计语言，其既具备传统编译型程序设计语言的强大功能，又在某种程度上具备比较简单的脚本和解析型程序设计语言的易用性。Python 语法简洁清晰，具有丰富和强大的类库，具有简单易学的面向对象的编程特点，同时还具备可移植、可扩展等特性，所以成为软件公司进行快速开发及科研单位进行科学研究的主流编程语言。它能够很轻松地将用其他语言（特别是 C/C++）编写的各种模块联结在一起。比如，先使用 Python 快速生成程序的原型，然后对其中有特别要求的部分，用更合适的其他语言改写。所以，Python 是一种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，具有很强的可读性。

Python 为我们提供了非常完善的基础代码库，覆盖了网络、文件、GUI（Graphical User Interface，图形用户界面）、数据库、文本等主要内容。用 Python 开发，许多功能不必从零开始编写，可以直接使用现成的模块。除了内置的基础库外，Python 还有大量的第三方库。许多大型网站就是用 Python 开发的，例如，国外著名的 YouTube 和 Instagram，还有国内的豆瓣，同时还包括 Google、Yahoo 等大型公司，甚至 NASA 都大量使用 Python。

## 1.2 Python 语言的发展历史

Python 语言的开发工作由 Guido van Rossum 开始于 1989 年末，接下来转移至荷兰的 CWI

(Centrum voor Wiskunde en Informatica, 国家数学和计算机科学研究院)并最终于1991年初公开发表。Guido van Rossum 是 CWI 的一名研究员,其认识到高级教学语言 ABC (All Basic Code) 因其语言不是开源的,不利于改进或扩展的重大缺点。因此,他决定开发一种可扩展的高级编程语言,为其研究小组的 Amoeba 分布式操作系统执行管理任务。他从 ABC 汲取了大量的语法,并从系统编程语言 Modular-3 借鉴了错误处理机制,开发出了一种能够通过类和编程接口进行扩展的高级编程语言,且将这种新语言命名为 Python (原意为大蟒蛇)——来源于 BBC (British Broadcasting Corporation, 英国广播公司) 当时热播的喜剧片 Monty Python。

自从1991年初公开发行后,Python 开发者和用户社区逐渐长大,Python 语言逐渐演变成一种成熟的、并获得良好支持的程序设计语言。2011年1月,Python 因在所有编程语言中占有最多市场份额,赢得 Tiobe2010 年度语言大奖。自从2004年以后,Python 的使用率是呈线性增长的趋势。

由于 Python 语言的简洁、易读以及可扩展性,所以在国外用 Python 做科学计算的研究机构日益增多,一些知名大学已经采用 Python 讲授程序设计课程,如卡耐基梅隆大学、麻省理工学院等。另外,众多开源的科学计算软件包都提供了 Python 的调用接口,如著名的计算机视觉库 OpenCV (Open Source Computer Vision, 开源计算机视觉库)、三维可视化库 VTK (Visualization Tool Kit)、医学图像处理库 ITK (Insight segmentation and registration Tool Kit)。而 Python 专用的科学计算扩展库就更多了,例如 NumPy、SciPy 和 Matplotlib 这3个十分经典的科学计算扩展库,都分别为 Python 提供了数值计算、科学计算以及绘图功能。因此,Python 语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表、甚至开发科学计算应用程序。

## 1.3 Python 语言的特点

Python 之所以具有较强的生命力是因为其拥有不同于(或优于)其他语言的特点,其主要特点如下。

### 1. 免费开源

Python 采取向公众开放源代码的策略,因而广大编程人员可以复制、阅读源代码并对其进行改进和完善,并运用到新的开源软件中。

### 2. 功能强大

Python 是能够进行系统调用的解析型脚本程序设计语言。它具有更高的数据结构,大大减少了项目中不可或缺的“程序框架”的开发时间。Python 语言还建立了更为有效的数据类型,比如列表和字典等,在减少开发时间的同时也缩短了代码长度。

### 3. 易学、易读

相对于其他语言,Python 语言关键字少,结构简单、语法清晰,具有很强的伪代码特性。没有其他语言通常用来定义变量、定义代码块等的命令式符号,使得 Python 代码方便阅读,初学者可以在更短时间内轻松上手。

### 4. 面向对象

Python 支持面向对象编程,同时还支持面向过程的编程。

### 5. 解释执行

Python 是一种解释型的语言,不需要编译成计算机可执行的二进制代码,可以直接从源代码

运行程序。Python 程序是通过 Python 解释器解释并执行的，Python 解释器把程序的源代码转换成称为字节码的中间形式，然后再将其翻译成计算机语言并执行，使得程序员无需关心程序如何编译、程序中用到的库如何加载等复杂问题。这样，使用 Python 将会更加简单，也更容易移植。

#### 6. 可扩展性

Python 的可扩展性使得程序员能够灵活地附加程序，缩短开发周期。因为 Python 是基于 C 语言开发的，所以一般用 C/C++ 来编写 Python 的扩展功能。

#### 7. 跨平台性

Python 具有强大的跨平台性和可移植性，只需要把 Python 程序复制到另一台计算机上就可以很方便地移植到各种主流的系统平台中。在任何一个平台上用 Python 开发的通用软件都可以稍做修改或者原封不动地在其他平台上运行。这种可移植性既适用于不同的架构，也适用于不同的操作系统。

Python 语言是跨平台的，它可以运行在 Windows、MAC 和各种 Linux/UNIX 系统上。在 Windows 上编写的 Python 程序，可以轻松方便地移植到 MAC 和各种 Linux/UNIX 系统上。

#### 8. 丰富的类库和内存管理器

Python 是世界上拥有最大标准库的编程语言。基于庞大的标准库，我们可以用 Python 编写程序来处理各种工作，包括文档生成、单元测试等功能。在 Python 的程序开发过程中，Python 解析器承担了程序的内存管理工作，使得程序员从内存事务处理中解脱出来，致力于程序功能的实现，从而减少错误，缩短开发周期。

## 1.4 Python 语言的应用

Python 的应用领域非常广泛，被普遍用于以下领域。

### 1. 系统编程

提供 API 编程接口，能够方便地进行系统维护和管理，是很多系统管理员理想的编程工具，是 Linux 系统下的标志性语言之一。

### 2. 图形处理

拥有庞大的对诸如 PIL、Tkinter 等图形类库的支持，能够方便地进行图形处理。

### 3. 数字处理

NumPy 扩展提供了大量与许多标准数学库对应的接口，可以方便地处理数学问题。

### 4. 文本处理

Python 提供了很多模块用于文本处理，如 re 模块能够处理正则表达式，XML 分析模块可进行文本的编程开发。

### 5. 数据库编程

通过 Python DB-API 规范模块，可以与 MS SQL Server、Oracle、Sybase、DB2、MySQL、SQLite 等数据库通信。Python 自带的 Gadfly 模块可提供完整的 SQL 环境。

### 6. 网络编程

提供丰富的模块支持 Socket 编程，能够方便、快速地开发分布式应用程序。

### 7. Web 编程

Python 包含了标准 Internet 模块，可用于实现各种网络任务。它也可使用大量的第三方工具

进行完整的、企业级的 Web 应用开发。

## 8. 多媒体应用

Python 的 PyGame 模块可用于编写游戏软件，同时，PyOpenGL 模块则封装了 OpenGL 应用程序编程接口，能进行二维和三维图象处理。

另外，Python 可广泛应用于科学计算、游戏、人工智能、机器人等领域。

# 1.5 Python 的安装

目前，Python 有两个版本系列，一个是 2.x 版，一个是 3.x 版。这两个版本是不兼容的。由于现在 Python 正在朝着 3.x 版本进化，在进化过程中，大量的针对 2.x 版本的代码要修改后才能运行。

## 1. Python 的获取

我们可以从 Python 的官方网站下载该软件。打开浏览器，在地址栏输入 <https://www.python.org/downloads/>，可以选择 2.x 系列的最新的 Python2.7.13 版本，也可以选择 3.x 系列的 Python3.4 版本，本书主要介绍 Python3.4 版本的使用。另外，软件还分 32 位和 64 位版本，本书选择 64 位版本。

## 2. Python 的安装

在下载目录中找到下载的 Python 安装文件 `python-3.4.4.amd64.msi`，双击这个文件，开始按安装向导进行安装。

## 3. 环境配置

把 Python 系统的安装目录，这里是“C:\Python34”，添加到系统环境变量 Path 中。

## 4. Python 的启动方式

当正确安装并配置了 Python 的环境变量后，就可以正常运行 Python 了。可以通过两种方式启动 Python：一种是使用命令行启动，另一种是使用 Python 的集成开发环境 IDLE 启动。

### (1) Python 的命令行启动

进入 DOS 命令行窗口，输入 `python`，出现如图 1-1 所示界面，则说明已成功在 DOS 系统下启动 Python 了。



图 1-1 DOS 系统下启动 Python

## (2) 使用 Python 集成开发环境 (IDLE) 启动

除了上面的方式, 我们还可以使用 Python 集成开发环境 (IDLE) 来启动 Python。单击“开始”菜单/“所有程序”/Python3.4/IDLE (Python 3.4 GUI-64bit) 启动 Python, 如图 1-2 所示。



图 1-2 使用集成开发环境 IDLE 启动 Python

## 1.6 第一个 Python 程序

接下来我们正式开始编写 Python 代码。首先, 我们通过 IDLE 启动解析器, 然后在交互模式下的提示符“>>>”下输入“print(“hello world”)”命令后回车, 即可看到 IDLE 窗口中会输出“hello world”, 如图 1-3 所示。

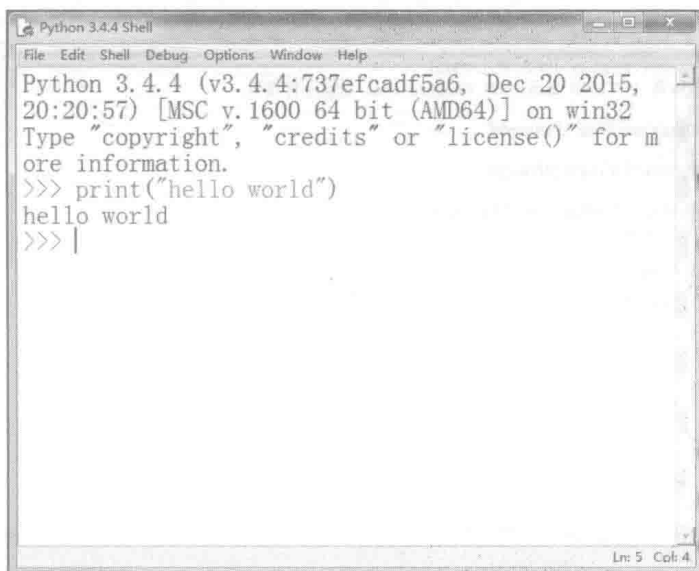


图 1-3 第一个 Python 程序

这行程序是调用了 Python 内置的一个 print 函数，在解析器上打印出引号里面的内容。

接下来讲解如何创建源文件编写上面的输出代码并调用该文件执行输出。

(1) 在 IDLE 下启动的 Python Shell 窗口中单击 File→New File，打开一个源文件编辑窗口。

(2) 在这里输入代码并保存为“FirstPython.py”，Python 源文件名以“.py”或“.pyw”为后缀名，如图 1-4 所示。



图 1-4 第一个 Python 程序的源文件

(3) 运行有两种方式。

① 在这个编辑器中单击 Run→Run Module 或按[F5]键运行程序，运行结果和图 1-3 类似，也显示在 IDLE 窗口中。

② 在 DOS 命令窗口进入到保存“FirstPython.py”文件所在的目录，输入“python FirstPython.py”并回车即可看到运行结果，如图 1-5 所示。



图 1-5 在命令行窗口下执行 Python 程序

## 小 结

本章首先讲解了什么是 Python 及 Python 语言的发展历史，然后介绍了 Python 语言的特点和应用，最后讲解了 Python 的安装并给出了第一个 Python 程序的创建、编辑和执行过程。

## 习 题

1. 安装 Python 并配置环境变量。
2. 在交互模式下用 print 函数输出“hello world”。
3. 创建一个 Python 的源文件，使用 print 函数输出你的基本信息，包括姓名、性别、年龄、住址等信息，并在 DOS 系统下执行该文件。



### 本章要点

- 理解 Python 程序的基本结构。
- 理解 Python 的输出格式。
- 掌握 Python 的基本数据类型。
- 掌握 Python 的输入输出函数。
- 掌握 Python 的运算符和表达式。

本章介绍 Python 程序的基本结构、简单数据类型、输入输出函数和 Python 的运算符和表达式。在介绍相关知识点的过程中结合例子进行说明，以便让读者更好地理解、掌握知识点。最后，本章末尾给出的练习题将使读者进一步巩固本章重要的知识点。

## 2.1 Python 程序基本结构

Python 程序结构涉及代码块、注释、语句分隔、语句续行、关键字与大小写等内容。

### 1. 用缩进表示代码块

在 Java、C/C++ 等语言中，用花括号表示代码块，Python 使用缩进（空格）来表示代码块。通常，语句末尾的冒号表示代码块的开始。在 if、for、while、函数、类等定义中都会使用到代码块。例如：

```
if x>100:
    y=x*5-1
else:
    y=0
```

在包含代码嵌套时，应注意同级代码块的缩进量保持相同（若不同，则可能会导致出错或得到意外的结果）。

### 2. 代码注释

Python 的代码注释有单行注释和多行注释，在运行程序时会忽略被注释的内容。单行注释用“#”表示注释开始，“#”之后的内容不会被执行。单行注释可以单独占一行，也可以放在语句末尾。多行注释是用 3 个英文的单引号“'''”或双引号“"""”作为注释的开始和结束符号。例如：

```
"""多行注释
下面的代码根据变量 x 的值计算 y
```