



普通高等教育“十三五”规划教材



# Python 3

## 程序设计学习指导与习题解答

高伟 李良俊 周应强 主编

 科学出版社

普通高等教育“十三五”规划教材

# Python 3 程序设计 学习指导与习题解答

高 伟 李良俊 周应强 主 编

刘德山 赵小平 副主编

胡 军 黎 谦 参 编

科 学 出 版 社

北 京

## 内 容 简 介

本书内容围绕全国计算机等级考试大纲编写,对二级 Python 语言程序设计考试大纲中的内容进行详细解析,深入挖掘。案例设计由浅入深,讲解详尽,突出应用性和指导性。以知识点、范例为基础,精心设计了习题与解答。本书配有面向自我学习提升和二级 Python 语言程序设计考试的模拟试题,力求提高学生独立解决问题的能力。

本书适合作为 Python 程序设计课程的学习指导教材,也可作为全国计算机等级考试二级 Python 语言程序设计考试的练习教材,还可供读者自学时参考。

### 图书在版编目(CIP)数据

Python 3 程序设计学习指导与习题解答 / 高伟, 李良俊, 周应强主编. — 北京: 科学出版社, 2019.9

(普通高等教育“十三五”规划教材)

ISBN 978-7-03-061801-6

I. ①P… II. ①高… ②李… ③周… III. ①软件工具-程序设计-高等学校-教学参考资料 IV. ①TP311.561

中国版本图书馆 CIP 数据核字 (2019) 第 131828 号

责任编辑: 宋 丽 吴超莉 / 责任校对: 王万红

责任印制: 吕春珉 / 封面设计: 东方人华平面设计部

科学出版社出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

三河市骏杰印刷有限公司印刷

科学出版社发行 各地新华书店经销

\*

2019年9月第一版 开本: 787×1092 1/16

2019年9月第一次印刷 印张: 14 1/2

字数: 340 000

定价: 38.00 元

(如有印装质量问题, 我社负责调换(骏杰))

销售部电话 010-62136230 编辑部电话 010-62135397-2052

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

# 前 言

本书是《Python 3 程序设计基础》(科学出版社, 刘德山、付彬彬、黄和主编)一书的配套教材, 是对主教材教学内容的补充和实践环节的完善。本书内容全面围绕全国计算机等级考试大纲编写, 对二级 Python 语言程序设计考试大纲中的内容进行详细解析, 深入挖掘。本书最后按照二级 Python 语言程序设计考试大纲的要求, 精心设计了 3 套模拟试卷。除第 13 章外, 每章均包括本章主要内容、典型例题分析、问题与思考、习题与解答 4 个部分。

## 1. 本章主要内容

以知识点的方式总结本章的主要内容, 包括基本概念、方法和应用。对于部分较难掌握的内容, 通过示例进行深入分析。

## 2. 典型例题分析

按照“源于主教材高于主教材”的原则对主教材中的示例进行扩展。对一部分基础、重要、易混淆的示例进行详细讲解; 对有代表性的和可扩展的示例, 给出由浅入深的解答, 有助于读者学习和理解; 对于部分较难理解的内容和因版本更新而增加的知识, 以示例的形式进行了补充。

## 3. 问题与思考

将基本和典型的知识点以问题的形式提出, 并给出解答。

## 4. 习题与解答

提供大量习题并附参考答案。读者完成习题后, 可以检查对课程内容的掌握程度, 对照参考答案, 可以发现问题, 有助于掌握所学知识。

本书由高伟、李良俊、周应强担任主编, 刘德山、赵小平担任副主编, 胡军、黎谦参与编写。其中, 第 1~4 章由高伟、周应强编写, 第 5~8 章由李良俊编写, 第 9~11 章由刘德山、赵小平编写, 第 12~13 章由胡军、黎谦编写, 附录由李良俊编写。书中的程序代码均运行通过, 部分程序有多种实现方法。

书中难免存在疏漏和不足之处, 欢迎读者和同行专家批评指正。

编 者

# 目 录

第 1 章 Python 概述	1
1.1 本章主要内容	1
1.2 典型例题分析	3
1.3 问题与思考	6
1.4 习题与解答	7
第 2 章 Python 基础知识	10
2.1 本章主要内容	10
2.2 典型例题分析	13
2.3 问题与思考	22
2.4 习题与解答	23
第 3 章 Python 程序的流程控制	31
3.1 本章主要内容	31
3.2 典型例题分析	33
3.3 问题与思考	42
3.4 习题与解答	42
第 4 章 Python 的组合数据类型	53
4.1 本章主要内容	53
4.2 典型例题分析	54
4.3 问题与思考	66
4.4 习题与解答	67
第 5 章 Python 函数	79
5.1 本章主要内容	79
5.2 典型例题分析	80
5.3 问题与思考	87
5.4 习题与解答	88
第 6 章 模块与 Python 的库	95
6.1 本章主要内容	95

6.2	典型例题分析	97
6.3	问题与思考	105
6.4	习题与解答	108
<b>第 7 章</b>	<b>Python 的文件操作</b>	<b>115</b>
7.1	本章主要内容	115
7.2	典型例题分析	118
7.3	问题与思考	121
7.4	习题与解答	122
<b>第 8 章</b>	<b>异常处理</b>	<b>129</b>
8.1	本章主要内容	129
8.2	典型例题分析	131
8.3	问题与思考	133
8.4	习题与解答	134
<b>第 9 章</b>	<b>tkinter GUI 编程</b>	<b>139</b>
9.1	本章主要内容	139
9.2	典型例题分析	141
9.3	问题与思考	146
9.4	习题与解答	147
<b>第 10 章</b>	<b>Python 的数据库编程</b>	<b>151</b>
10.1	本章主要内容	151
10.2	典型例题分析	153
10.3	问题与思考	157
10.4	习题与解答	157
<b>第 11 章</b>	<b>面向对象程序设计</b>	<b>163</b>
11.1	本章主要内容	163
11.2	典型例题分析	165
11.3	问题与思考	172
11.4	习题与解答	173
<b>第 12 章</b>	<b>Python 的第三方库</b>	<b>178</b>
12.1	本章主要内容	178
12.2	典型例题分析	181
12.3	问题与思考	185
12.4	习题与解答	187

第 13 章 模拟测试 .....	191
13.1 模拟试卷 1 .....	191
13.2 模拟试卷 2 .....	202
13.3 模拟试卷 3 .....	212
附录 全国计算机等级考试二级 Python 语言程序设计考试大纲（2018 年版） .....	222
参考文献 .....	224



# 第1章

## Python 概述

### 1.1 本章主要内容

本章重点介绍计算机语言的基础知识和 Python 程序设计的基础知识，内容与全国计算机等级考试二级 Python 语言程序设计考试大纲一致。

#### 1. 计算机语言

计算机语言的发展经历了机器语言、汇编语言和高级语言等阶段。

机器语言是采用计算机指令格式并以二进制编码表达各种操作的语言。机器语言的特点是能够被计算机直接识别，执行速度快，占用存储空间小。

汇编语言是一种符号语言，它用助记符来表达指令功能。

机器语言和汇编语言统称为低级语言。

高级语言是面向问题的语言，比较接近于人类的自然语言。高级语言是与计算机结构无关的程序设计语言，可以方便地表示数据的运算和程序控制结构，能更有效地描述各种算法，使用户容易掌握和理解。

Python 是一种高级语言。

#### 2. 编译与解释

计算机程序的执行方式可分为编译和解释两种。

编译是将源程序代码转换成目标代码的过程。源代码通常是计算机高级语言代码，而目标代码则是机器语言代码。执行编译的计算机程序称为编译器（compiler）。

解释是将源代码逐条转换成目标代码，同时逐条运行目标代码的过程。执行解释的计算机程序称为解释器（interpreter）。

编译和解释的区别在于编译是一次性地翻译，程序被编译后，运行时不再需要源代码；解释则在每次程序运行时都需要解释器和源代码。

Python 是解释型语言。

### 3. Python 的版本

Python 的发展过程中，存在 Python 2.x 和 Python 3.x 两个不同系列的版本，两个版本之间不兼容。目前是 Python 2.x 和 Python 3.x 两个版本并存。Python 2.x 的最高版本是 Python 2.7，Python 3.x 从 2008 年开始发布。

Python 2.x 和 Python 3.x 两个不同版本并存的原因是，Python 3.0 发布时不支持 Python 2.0，导致很多用户无法正常升级使用 Python 3.0，所以后来又发布了 Python 2.7，Python 2.7 将会被支持到 2020 年。

### 4. Python 的开发环境

Python 是一个轻量级的软件，其安装包可以在 Python 官网下载。Python 开发包自带的编辑器 IDLE 是一个集成开发环境（integrated development environment, IDE），启动文件是 idle.bat，它位于安装目录的 Lib\idlelib 文件夹下。

IDLE 的常用快捷键如下：

- Ctrl + [：缩进代码。
- Ctrl + ]：取消缩进代码。
- Alt+3：注释代码行。
- Alt+4：取消注释代码行。
- Alt+/: 单词自动补全。
- Alt+P：浏览历史命令（上一条）。
- Alt+N：浏览历史命令（下一条）。
- F5：运行程序。
- Ctrl+F6：重启 Shell，之前定义的对象和导入的模块全部失效。

PyCharm 是 JetBrains 公司开发的一款专业级的 Python IDE，PyCharm 具有典型 IDE 的多种功能。用户可以根据自己的操作系统平台下载不同版本的 PyCharm，并且每个平台可以选择下载 Professional 和 Community 两个版本。PyCharm Professional 是需要付费的版本。

### 5. Python 程序执行原理

Python 是一种解释执行的脚本语言，可以直接运行。从计算机的角度看，Python 程序的运行过程包含两个步骤：解释器将源代码翻译成字节码，然后由虚拟机运行解释执行，如图 1-1 所示。

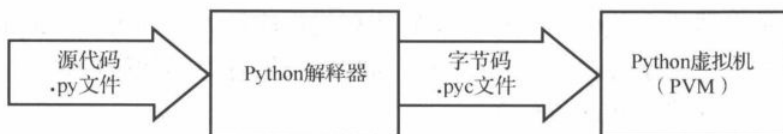


图 1-1 Python 程序的执行过程

Python 代码源文件的扩展名通常为.py。在执行时，首先由 Python 解释器将.py 文件中的源代码翻译成中间代码（中间代码是一个扩展名为.pyc 的文件），再由 Python 虚拟机（Python virtual machine, PVM）逐条将字节码翻译成机器指令执行。

## 6. 程序设计方法

IPO 模式是一种典型的程序设计模式，即程序包括输入（input）、处理（process）、输出（output）3 部分。输入是程序设计的起点，包括文件输入、网络输入、交互输入、参数输入等；输出是程序展示运算成果的方式，包括文件输出、网络输出、控制台输出、图表输出等；而处理部分则是编程的核心，包括数据处理与赋值，更重要的是算法。例如，给定两点的坐标，求两点之间的距离，需要一个公式，这个公式就是一个算法；再如，求三角形面积的海伦公式也是一个算法。更多的算法需要用户去设计，如从一组数据中查找某一数据的位置，这就需要根据数据的特点，由用户设计算法。

除了 IPO 模式外，通过添加足够的注释来增强程序的可读性、通过调试来完善程序，都是程序设计中不可缺少的环节。

## 1.2 典型例题分析

【1】阅读下面的代码，逐行解析程序。

```
01 #demo0101.py
02 '''
03 计算圆锥体体积，V=1/3*s*h
04 完成时间：20190201
05 '''
06 import math
07 r=eval(input("请输入圆锥体的半径："))
08 h=eval(input("请输入圆锥体的高："))
09 v=math.pi*r*r*h/3      #计算圆锥体体积的公式
10 print("圆锥体体积为{:.2f}".format(v))
```

解析：

（1）第 1 行～第 5 行是注释，其中第 1 行是单行注释；第 2 行～第 5 行由 3 个单引号（或 3 个双引号）包围的是多行注释，多用于大量信息描述，一般用于说明程序的功能。

（2）第 6 行是导入语句，用于导入 math 模块，使得第 9 行的 math.pi 可以获得具体的圆周率值。

（3）第 7 行和第 8 行包括两个函数 input()和 eval()，其中 input()函数用于接收用户的输入，接收的值是字符串；eval()函数用来执行一个字符串表达式，并返回该表达式的值。

（4）第 9 行是程序的核心，利用公式计算圆锥体的体积；第 10 行是输出语句。

该程序保存在扩展名为.py 的文件中，本例中文件名为 demo0101.py，编译后产生字节码文件（扩展名为.pyc），如果在 IDLE 下直接运行文件，该字节码文件不可见。程序执行到第 7 行、第 8 行后暂停，等待接收用户输入，之后输出。

按程序设计的 IPO 模式，程序接收键盘输入的半径和高，第 9 行是数据处理，计算圆锥体体积，最后输出。这是一个典型的 IPO 模式的程序。

**【2】** 阅读下面的程序，说明程序的功能。

```
01 #demo0102.py
02 words=input("请输入 3 个英文单词，用逗号分隔：")
03 a,b,c=words.split(",")
04 if a>b:
05     a,b=b,a
06 if a>c:
07     a,c=c,a
08 if b>c:
09     b,c=c,b
10 print("按字典序的 3 个单词是",a,b,c)
```

**解析：**

(1) 程序的功能是对 3 个输入的单词按字典序排序，第 2 行要求使用一个 input() 函数直接输入 3 个单词。

(2) 第 3 行使用字符串的 split() 方法拆分字符串，拆分后的 3 个单词赋给变量 a、b、c。

(3) 第 4 行~第 9 行是字符串比较的算法，第 10 行是输出语句。

请读者注意体会使用 split() 方法后，程序更为简洁。可以使用列表这种数据结构来进一步修改程序。

```
words=input("请输入英文单词，用逗号分隔：")
lst=words.split(",")
lst.sort()
print("按字典序的结果是",lst)
```

上面代码中，使用字符串的 split() 方法将单词解析到列表 lst 中，然后调用列表的 sort() 方法，完成列表中单词的排序。实际上这段代码更适合多个单词的排序。也可以使用下面的方法按降序排序。

```
lst.sort(reverse=True)
```

请读者到 Python 帮助文档中查阅列表的 sort() 方法相关参数的使用说明。

**【3】** 查阅 Python 帮助文档，查找其中的 Numeric Types 类型，试使用其中的代数函数、指数和对数函数、三角函数等完成计算功能。

**解析：**

Python 帮助文档提供了语言参考及标准模块的详细信息，是学习和使用 Python 的必备工具。

在 IDLE 环境下, 选择[Help]/[Python Docs]命令或按 F1 键, 就可以启动 Python 帮助文档, 如图 1-2 所示。

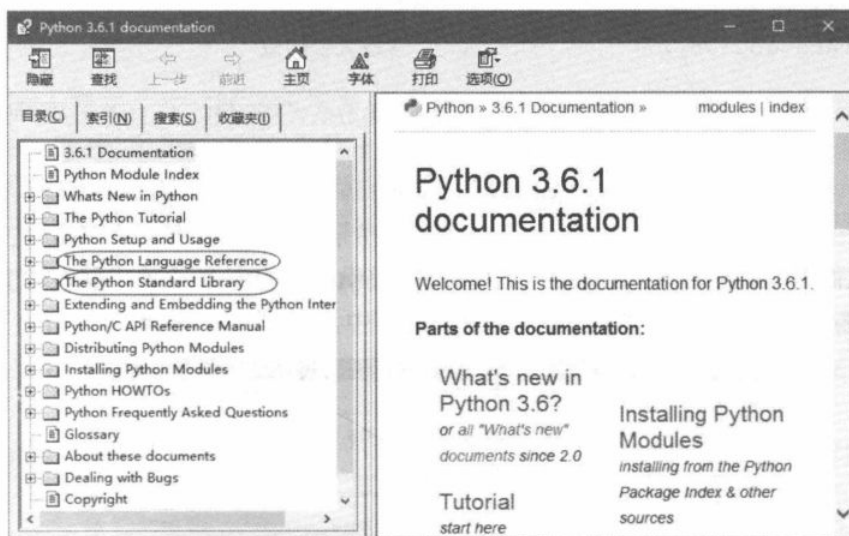


图 1-2 Python 帮助文档初始界面

(1) 本题中的 Numeric Types 是数值类型, 包括整型、浮点型和复数型 3 种。在帮助文档的“[The Python Standard Library](#)”选项下, 选择“[Numeric and Mathematical Modules](#)”模块, 其中包括数学运算的相关函数, 如图 1-3 所示。

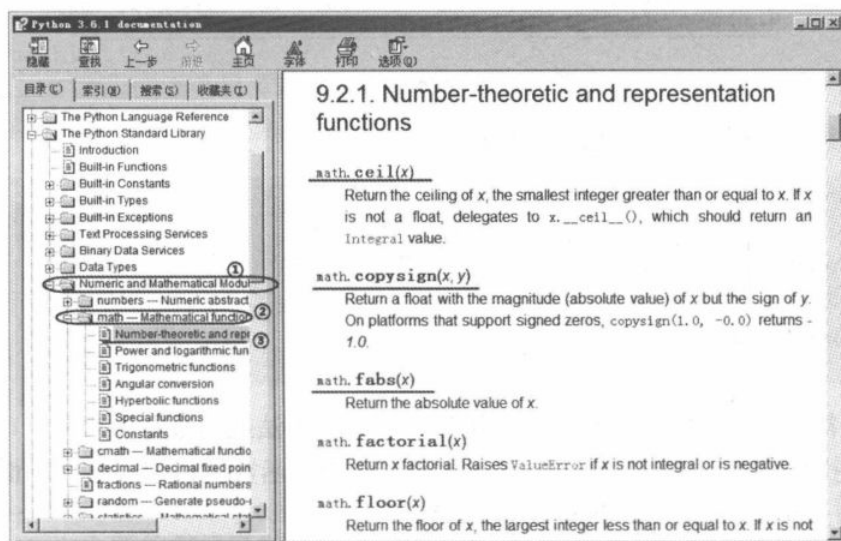


图 1-3 查找 math 模块中的函数

(2) 要查找数学函数的使用方法, 可以按图 1-3 所示的步骤查阅文档, 还可以使用关键词查找具体函数的使用方法。

(3) 如果要使用函数库中一些函数的具体计算功能, 因为是 math 库中的函数, 所以需要先导入 math 库, 举例如下:

```

>>> import math
>>> math.fsum((1,2,4.55,-56))      #求和，数据需要在元组或列表中
-48.45
>>> math.gcd(36,27)               #最大公约数
9
>>> math.pow(13,2)                #指数函数
169.0
>>> math.sqrt(3)                  #平方根函数
1.7320508075688772
    
```

(4) 上面是 math 库中的函数，也可以选择 Python 的内置函数，其在“The Python Standard Library”选项下的“Built-in Functions”中，如图 1-4 所示。

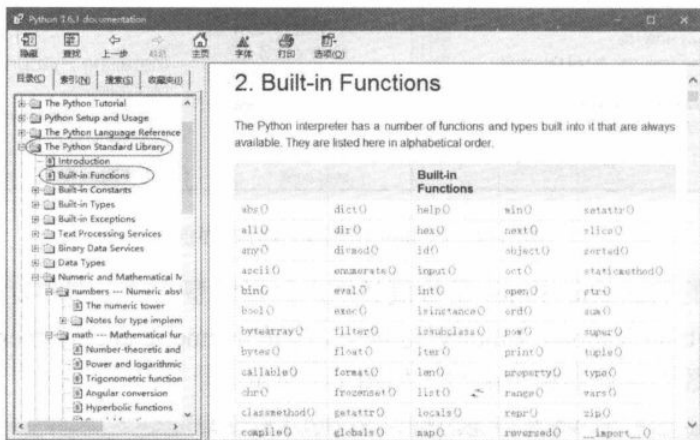


图 1-4 Python 的内置函数

内置函数不需要导入，可以直接使用，举例如下：

```

>>> max(45,21,-99,34.28)
45
>>> sum((14,201,-5.9,34.28))
243.38
>>> len("student")
7
    
```

Python 帮助文档通常是英文文档，初学者在理解语法或函数有困难时，可以借助搜索引擎来查找知识点，也可以借助“菜鸟教程”来查找函数的使用方法。Python 菜鸟教程的网址是：<http://www.runoob.com/python3>。

## 1.3 问题与思考

### 一、问题

1. Python 源文件的扩展名是什么？编译后生成的字节码文件的扩展名是什么？该

字节码文件保存在什么位置?

2. Python 解释器环境中的特殊变量 “\_” (单下画线) 有什么用途?

## 二、解答

1. Python 源文件的扩展名通常为.py。

程序在执行时, Python 解释器将.py 文件中的源代码翻译成扩展名为.pyc 的字节码文件, 再由 Python 虚拟机逐条将字节码翻译成机器指令执行。

.pyc 文件保存在 Python 安装目录的\_\_pycache\_\_文件夹下, 如果 Python 无法在用户的计算机上写入字节码, 字节码文件将只在内存中生成, 并在程序结束运行时自动丢弃。主文件(直接执行的 Python 程序文件)因为只需要装载一次, 所以并没有保存.pyc 文件。当 Python 源文件用于 import 导入时, 将会生成.pyc 文件, 并且在\_\_pycache\_\_文件夹下可以找到该文件。

2. Python 解释器环境中存在一个特殊变量 “\_”, 用于表示上一次运算的结果。使用该变量, 可以更方便用户调试程序。例如:

```
>>> import math
>>> r=2      #柱体半径
>>> h=4      #柱体高
>>> math.pi*r*r
12.566370614359172
>>> _       #上次运行结果
12.566370614359172
>>> math.pi*r*r
>>> _*h     #计算体积
50.26548245743669
```

## 1.4 习题与解答

### 一、习题

#### 1. 选择题

(1) Python 属于 ( )。

- A. 机器语言    B. 汇编语言    C. 高级语言    D. 以上都不是

(2) Python 内置的集成开发环境是 ( )。

- A. PyCharm    B. Pydev    C. IDLE    D. pipy

(3) 关于 Python 的特点, 以下选项中描述错误的是 ( )。

- A. Python 是脚本语言    B. Python 是非开源语言  
C. Python 是跨平台语言    D. Python 是多模型语言

- (4) IDLE 中, 将选中区域注释的组合键是 ( )。
- A. Alt+G      B. Alt+3      C. Alt+4      D. Alt+Z
- (5) 以下选项中, 不是 Python IDE 的是 ( )。
- A. PyCharm      B. Jupyter Notebook  
C. R Studio      D. Spyder
- (6) IDLE 中, 将选中区域缩进的组合键是 ( )。
- A. Ctrl+S      B. Ctrl+[      C. Ctrl+A      D. Ctrl+C
- (7) 下列关于 Python 版本的选项中, 正确的是 ( )。
- A. 目前存在的 Python 3.x 版本兼容 Python 2.x 版本的程序  
B. Python 2.x 版本需要升级到 Python 3.x 版本才能使用  
C. 目前的 Python 2.x 版本已经被淘汰  
D. Python 2.x 和 Python 3.x 是两个不兼容的版本
- (8) Python 解释器有多种计算机语言实现的版本, 下面选项中不是 Python 解释器实现的是 ( )。
- A. Cpython      B. Jpython      C. IronPython      D. MPython
- (9) 在 Python 函数中, 用于获取用户输入的是 ( )。
- A. input()      B. print()      C. get()      D. eval()
- (10) 以下选项中, 不是 IPO 模型一部分的是 ( )。
- A. Input      B. Output      C. Program      D. Process

## 2. 编程题

(1) 从键盘输入存款金额和存款年限 (n), 按“金额 $\times(1+\text{利率})^n$ ”计算收益。其中, 默认利率为 5.2%。

(2) 完善上面的程序, 当存款年限大于或等于 3 年时, 利率上浮 20%。

(3) 交互输入三角形的两边长和夹角, 计算第三边的边长。

## 二、参考答案

### 1. 选择题

(1) ~ (5) CCBBC      (6) ~ (10) BDDAC

### 2. 编程题

(1)

```
num=eval(input("请输入存款金额: "))
years=eval(input("请输入存款年限: "))
rate=0.052
total=num*(1+rate)**years
print("最终收益为: ",total)
```

(2)

```
num=eval(input("请输入存款金额: "))
years=eval(input("请输入存款年限: "))
rate=0.052

if years>=3:
    rate=rate*(1+0.2)
total=num*(1+rate)**years
print("最终收益为: {:.2f}".format(total));
```

(3)

```
import math
data=input("请输入两边长及夹角, 用逗号分隔: ")
a,b,theta=data.split(",")
a=float(a)
b=float(b)
theta=float(theta)
c=math.sqrt(a**2+b**2-2*a*b*math.cos(theta*math.pi/180))
print(c)
```



## 第 2 章

# Python 基础知识

### 2.1 本章主要内容

本章重点介绍 Python 作为计算机程序设计语言编程的基础知识，内容包括 Python 的基本语法。通过学习本章，读者能够正确运用不同类型的数据和运算符编写符合规范的程序，掌握程序设计的思想和基本方法。

本章是 Python 学习的重点之一，内容与全国计算机等级考试二级 Python 语言程序设计考试大纲一致。

#### 1. Python 程序的书写规范

Python 通常是一行书写一条语句，如果一行内写多条语句，要求使用分号分隔。如果一条语句过长，可以使用反斜线“\”来实现分行书写功能。在()、[]、{}内的跨行语句，也被视为一行语句。

代码块是用于完成一个相对完整功能的多行代码，如 if、while 等语句以冒号(:) 结束，该语句行之后的一行或多行代码构成代码块。与其他一些程序设计语言，如 C 语言、Java 语言等使用大括号{}表示代码块不同，Python 中的代码块使用缩进来表示，即要求同一个代码块的每行语句前必须包含相同的缩进空格数，一般为 4 个空格。

在程序中添加适当的注释是十分必要的。注释的主要目的是阐述代码要做什么，以及是如何做的。注释以自然语言对问题的解决方案加以描述，增加程序的可读性。Python 的注释分为单行注释和多行注释两种，其中单行注释以“#”开头，多行注释使用 3 个引号作为开始和结束符号。程序执行时，注释中的内容会被 Python 解释器忽略。

#### 2. Python 的变量

变量是指向特定值的名称。变量一般用于存储程序中的数据。程序中可以随时修改