



高等院校计算机类规划教材

全国高等院校计算机基础教育研究会重点立项项目



# Python

# 程序设计基础

主 编 顾鸿虹 于 静

副主编 陈儒敏 顾玲芳

参 编 杨 娜 张 虹 冯 瑶 沈加锐



北京邮电大学出版社  
www.buptpress.com



高等院校计算机类规划教材

全国高等院校计算机基础教育研究会重点立项项目

# Python 程序设计基础

主 编 顾鸿虹 于 静

副主编 陈儒敏 顾玲芳

参 编 杨 娜 张 虹 冯 瑶 沈加锐



北京邮电大学出版社  
[www.buptpress.com](http://www.buptpress.com)

## 内 容 简 介

本书从程序设计的基础概念出发,基于 Windows 系统和 Python 3.6 搭建程序开发环境,通过编写 Python 程序代码讲解程序设计的相关内容,强调计算思维的培养。全书共 6 章,内容包括程序设计与程序设计语言、Python 基础、程序结构控制、函数与模块、turtle 库的应用和文件处理。书中各章将所涉及的知识与相应示例代码有机结合,注重应用实践。本书在附录中提供了全国计算机等级考试(NCRE)二级的 Python 语言程序设计科目的模拟题。

本书内容由浅入深,循序渐进,同时本书为读者提供了丰富的程序案例。本书可作为高等院校,特别是应用型本科院校程序设计基础课程的教学用书,也可作为程序设计初学者或是对 Python 感兴趣的自学者的参考教程。

### 图书在版编目(CIP)数据

Python 程序设计基础 / 顾鸿虹, 于静主编. -- 北京: 北京邮电大学出版社, 2020. 4  
ISBN 978-7-5635-6032-5

I. ①P… II. ①顾… ②于… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2020)第 055243 号

策划编辑: 马晓仟 责任编辑: 徐振华 王小莹 封面设计: 七星博纳

---

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷:

开 本: 787 mm×1 092 mm 1/16

印 张: 10

字 数: 235 千字

版 次: 2020 年 4 月第 1 版

印 次: 2020 年 4 月第 1 次印刷

---

ISBN 978-7-5635-6032-5

定价: 26.00 元

· 如有印装质量问题, 请与北京邮电大学出版社发行部联系 ·

# 前 言

随着新工科建设要求的提出,以及大数据和人工智能应用研究热潮的兴起,Python 程序设计语言因其简单易学、易用、易维护且功能强大,在大数据和人工智能研究领域被广泛应用,同时也被广泛应用于各种应用程序的开发。Python 语言是一种开源的解释型高级程序设计语言,支持面向对象,具有丰富强大的库,并且能够与多种程序设计语言完美融合。

本书从程序设计的基础概念出发,基于 Windows 系统和 Python 3.6 搭建程序开发环境,通过编写 Python 程序代码讲解程序设计的相关内容,将各知识点与相应示例代码有机结合,由浅入深,循序渐进。书中具有丰富的程序案例及编程思路的解析,有利于对程序的理解和计算思维的培养。

全书共包含 6 章,具体内容如下。

第 1 章是程序设计与程序设计语言,介绍了程序、程序设计及程序设计语言的概念,讲解了常用程序的设计方法(IPO 程序设计方法)和算法流程图的绘制,简单介绍了 Python 的发展和特点,详细介绍了 Python 开发环境的安装与配置以及使用 Python 自带 IDLE 编写和运行 Python 程序的方法。

第 2 章是 Python 基础,介绍了 Python 程序的格式框架,包括辅助性信息、缩进和续行等;详细讲解了变量与变量的赋值;重点介绍了 Python 中所支持的简单数据类型(数字型、非数字型)、Python 中不同数据类型的数据运算处理方法,以及不同数据的格式化输出方法;介绍了组合数据类型数据及其操作方法。

第 3 章是程序结构控制,介绍了程序的三种基本结构——顺序结构、分支结构和循环结构;详细介绍了三种程序结构的特点和语法结构;通过分析例题的详细解题思路重点讲解程序设计方法;介绍了循环中 continue、break 和 else 语句的使用方法;介绍了 Python 程序中进行异常处理的方法。

第 4 章是函数与模块,首先介绍了使用函数的意义,然后介绍了 Python 语言中定义函数的方法以及调用函数的方法,其中详细地讲解了函数的参数和函数的返回值,最后由函数提出了变量作用域的相关知识并提供了函数应用的简单示例。本章在函数的基础上,进一步介绍了通过模块封装程序的方法,并详细介绍了 random 标准库的使用。除了标准库外,第 4 章还介绍了第三方库的安装方法和使用方法,包的相关概念以及搜索路径的配置。

第5章是turtle库的应用,详细介绍了turtle库的使用,通过丰富的示例代码讲解了各种常用函数的使用方法,并通过使用turtle绘图的案例帮助读者建立程序设计的逻辑思维。

第6章是文件处理,介绍了使用Python语言操作文本文件的常用函数〔如open()、read()、readline()和readlines()等〕,介绍了文件处理中常用的os模块和os.path模块。

本书的附录I列举了Python中的标准异常类型,附录II提供了丰富的全国计算机等级考试二级模拟题。

本书第1章由顾鸿虹、沈加锐编写,第2章由陈儒敏、顾玲芳编写,第3章、附录I由杨娜编写,第4章由冯瑶、顾鸿虹编写,第5章由张虹编写,第6章由于静编写,附录II由于静、陈儒敏编写。全书由顾鸿虹负责内容结构设计和统稿工作。

由于编者水平有限,书中难免有疏漏之处,恳望读者批评指正。

编 者

# 目 录

<b>第 1 章 程序设计与程序设计语言</b> .....	1
1.1 概述 .....	1
1.1.1 程序定义 .....	1
1.1.2 程序设计语言 .....	2
1.1.3 程序设计方法 .....	4
1.2 Python 简介 .....	7
1.2.1 Python 的发展 .....	7
1.2.2 Python 的特点 .....	8
1.3 Python 环境安装与配置 .....	9
1.3.1 Python 环境安装 .....	9
1.3.2 添加环境变量.....	12
1.3.3 Python 程序的编写与运行 .....	13
习题 .....	16
<b>第 2 章 Python 基础</b> .....	17
2.1 Python 程序格式框架 .....	17
2.1.1 辅助性信息.....	18
2.1.2 缩进.....	18
2.1.3 续行.....	19
2.2 变量与变量的赋值.....	20
2.3 基本数据类型.....	21

2.3.1	数字类型	21
2.3.2	非数字类型	22
2.3.3	变量数据类型查看及类型转换	24
2.3.4	input()函数与eval()函数	26
2.4	数值运算	27
2.5	print()函数与格式化输出	34
2.5.1	格式化浮点数输出	34
2.5.2	格式化整数输出	36
2.5.3	格式化字符串输出	37
2.5.4	f-string 格式化	38
2.6	组合数据类型	40
2.6.1	字符串	40
2.6.2	列表	45
2.6.3	元组	50
2.6.4	字典	52
2.6.5	集合	55
	习题	58
<b>第3章</b>	<b>程序结构控制</b>	<b>60</b>
3.1	顺序结构	61
3.2	分支结构	62
3.2.1	单分支语句	62
3.2.2	双分支语句	64
3.2.3	多分支语句	65
3.2.4	分支嵌套	67
3.3	循环结构	69
3.3.1	while 语句	69
3.3.2	for 语句	71
3.3.3	break 语句和 continue 语句	73

3.3.4 循环嵌套·····	74
3.3.5 循环语句中 else 的使用·····	77
3.4 异常处理·····	78
3.4.1 try···except 语句·····	78
3.4.2 try···finally 语句·····	80
3.5 综合应用·····	81
习题·····	86
<b>第 4 章 函数与模块</b> ·····	<b>87</b>
4.1 函数·····	88
4.1.1 函数的定义与调用·····	89
4.1.2 函数的参数·····	90
4.1.3 函数的返回值·····	93
4.1.4 变量的作用域·····	93
4.1.5 函数应用·····	96
4.2 模块·····	99
4.2.1 模块的导入·····	99
4.2.2 random 标准库·····	101
4.2.3 第三方库·····	106
4.2.4 包·····	107
4.2.5 搜索路径·····	108
习题·····	109
<b>第 5 章 turtle 库的应用</b> ·····	<b>110</b>
5.1 turtle 常用函数·····	110
5.2 使用 turtle 绘制图形·····	116
习题·····	123
<b>第 6 章 文件处理</b> ·····	<b>124</b>
6.1 文件基础操作·····	124

6.1.1 文件的打开与关闭 .....	125
6.1.2 文件的读写 .....	126
6.1.3 使用 with 打开文件 .....	130
6.2 os 模块及 os.path 模块 .....	131
6.2.1 os 模块 .....	131
6.2.2 os.path 模块 .....	133
6.3 文件读写应用 .....	134
习题 .....	138
参考文献 .....	139
附录 I Python 标准异常 .....	140
附录 II 全国计算机等级考试二级模拟题 .....	142

# 第 1 章 程序设计与程序设计语言

---

## 本章要点

- 计算机程序的概念。
- 程序设计方法: IPO 程序设计方法和算法流程图。
- Python 开发环境的安装与使用。
- Python 程序的编写与运行。

## 1.1 概 述

计算机程序设计是以某种程序设计语言为工具, 给出解决某一特定问题的计算机程序的过程。

### 1.1.1 程序定义

现代汉语词典里对“程序”的解释是事情进行的先后次序。我国 2016 版国家标准《质量管理体系基础和术语》中对于“程序”的定义是为进行某项活动或过程所规定的途径。上述不论是哪一种解释, 都蕴涵着为完成某件事情而要经历的方法流程。而本书中所说的程序则专指计算机程序。计算机俗称电脑, 是一种能够自动、高速处理海量数据的现代化智能电子设备, 具有存储记忆功能。计算机之所以能够自动处理数据正是因为其中存储了相应的控制程序, 使用时通过程序指挥计算机的各个部分协同工作完成数据处理工作。

**【例 1-1】** turtle 绘图演示。

```
# 1-1. py
from turtle import *

setup(840,500)
speed(5)
pensize(4)
hideturtle()
colormode(255)
color(255,155,192)
penup()
```

```
goto(-69,167)
pendown()
begin_fill()
setheading(180)
circle(300,-30)
circle(100,-60)
circle(80,-100)
circle(150,-20)
circle(60,-95)
setheading(161)
circle(-300,15)
end_fill()
color(239,69,19)
penup()
goto(-20,30)
pendown()
setheading(-80)
circle(30,40)
circle(40,80)
done()
```

将例题中的程序运行起来,会看到有“画笔”在屏幕中自动绘图,而这支画笔的动作就是由上述计算机程序来指挥控制实施的,你能对照运行过程猜出上述程序中的每一行在做什么吗?

所谓计算机程序,是指使用特定语言编写的,运行在计算机上的一组能够指挥计算机完成某种工作指令的集合。例 1-1 是一段用 Python 语言编写的计算机程序。

### 1.1.2 程序设计语言

程序设计是给出解决特定问题的程序的过程。程序需要使用某种特定的程序设计语言作为工具进行编写,因此程序设计的學習必须借助于一种程序设计语言,例 1-1 所使用的 Python 语言即为本书所采用的程序设计语言。

程序设计语言是用于与计算机进行交互(交流)的人造语言,也称编程语言。程序设计语言由一组符号和特定的规则组成,人类将这些符号按照相应的规则组织起来,形成计算机能够理解的指令,指挥计算机工作。程序设计语言比自然语言更简单、更严谨、更精确。历史上出现的程序设计语言有上千种,这些程序设计语言被分为 3 大类:机器语言、汇编语言和高级语言。

#### 1. 机器语言

机器语言使用二进制代码表示机器指令,是计算机能够直接识别和执行的一种程序设计语言。使用机器语言时,程序员们将用 0、1 数字编成的程序代码打在纸带或卡片上

(1 打孔,0 不打孔),再将程序通过纸带机或卡片机输入计算机进行运算处理。因为计算机能够直接执行二进制指令,所以机器语言是运行效率最高的程序设计语言,但是机器语言使用 0、1 二进制代码表示,使得编写、阅读和修改机器语言代码变得十分困难,而且机器语言与硬件关联紧密,不同型号的计算机其机器语言不能通用,因此,除特别需求外,如今的编程人员不会学习机器语言。

## 2. 汇编语言

汇编语言是一种用于电子计算机、微处理器、微控制器或其他可编程器件的低级语言,亦称为符号语言,它使用助记符代替机器指令的操作码,用地址符号代替指令或操作数的地址。不同的设备有对应不同机器语言的一整套指令,称为指令集。执行使用汇编指令编写的程序时,需要先通过汇编编译器将汇编指令转换成机器指令,然后指挥计算机完成相应的操作。汇编语言在一定程度上改善了程序的可读性,但开发效率依然较低,它和机器语言一样是面向机器的低级语言,所编写的程序仍然缺乏可移植性,也不易维护。

在实际应用中,汇编语言通常被应用在底层,包括硬件操作和有高时效性要求的程序,如驱动程序、嵌入式操作系统和实时运行程序。

## 3. 高级语言

高级语言是相对于低级语言而言的,是高度封装的程序设计语言,与计算机的硬件结构及指令系统无关,更接近于人类的自然语言,这使程序的编写更容易,可读性更高,更加易于学习。目前流行的高级编程语言多是基于英语的,20 世纪 80 年代开始,我国以沈志斌为代表的研究人员也积极开发汉语程序设计语言,如汉编、易语言和习语言等,但近年来一直未见推广,也未再进行更新升级。

高级语言并不是特指的某一种具体的语言,而是包括很多编程语言,如流行的 C、Java、C++、Python 等。高级语言所编写的程序称之为源程序,源程序不能被计算机识别,必须转换成机器语言才能被执行,按照转换方式的不同可将高级语言分为两类:解释型语言和编译型语言。

### (1) 解释型语言

解释型语言编写的源程序在运行时,需要由专门的解释器将源代码逐条解释,边解释边执行,如图 1.1 所示。

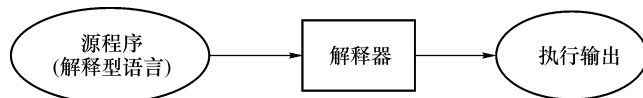


图 1.1 解释执行过程

解释型语言源程序不能生成可独立执行的可执行文件,不能脱离解释器,每执行一次都要翻译一次,因此效率比较低,但可以动态地调整、修改源程序,比较灵活。Python 语言属于解释型语言。

### (2) 编译型语言

编译型语言在源程序执行之前,先将源代码一次性整体编译成目标代码(机器语言)文件,执行时,直接执行目标代码即可,如图 1.2 所示。

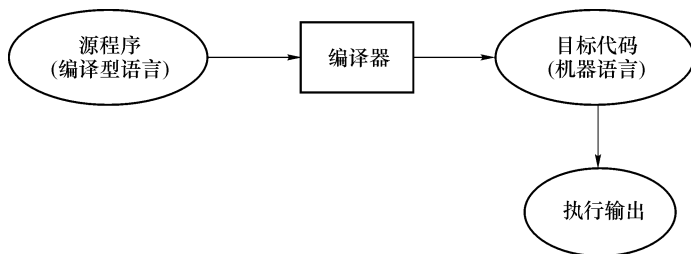


图 1.2 编译执行过程

目标程序可以脱离源程序独立执行,使用比较方便、效率较高,但应用程序一旦需要修改,则必须先修改源代码,再重新编译生成新的目标文件后才能执行,如果没有源代码,将无法修改。现在大多数的高级语言都是编译型语言,如 C、C++、Java 等。

虽然历史上出现了上千种程序设计语言,但大多语言由于应用领域狭窄或兼容性差等原因导致生命力不够强劲,已经停用,C 语言是第一个被广泛使用的编程语言,直到今天也还在使用。而 Python 语言自从 20 世纪 90 年代初诞生至今,特别是在经历过版本升级之后,因其简洁性、易学易用性、可扩展性以及与其他流程序序设计语言程序的易结合性,迅速成为最流行、最好用的编程语言之一。

### 1.1.3 程序设计方法

程序设计的过程是在提出问题后,思考解决问题的流程,也是通过在计算机上运行某种程序设计语言编写成的程序来解决问题的过程,也就是说,程序设计应当包括问题分析、算法设计、程序编写、程序调试和升级维护五个阶段。

算法设计是程序设计的核心,算法即指解决问题的方法步骤。做任何事情都有一定的步骤。例如,在数学运算中要遵循先乘除后加减的步骤;生活中要乘坐火车时,首先要购买车票,然后按时到达火车站取票,检票上车。解决问题的步骤都是按照一定的顺序进行的,当遇到的问题规模越庞大时,步骤也就越复杂,这时就需要通过一些方式把步骤描述记录下来,以便后续按照步骤执行。

为了描述记录一个算法,可以用不同的方式。常用的方式有自然语言、流程图、伪代码、PAD 图等,其中较为常用的是传统算法流程图。在传统算法流程图中使用 ANSI (美国国家标准化协会)规定的标准符号来表示各种类型的操作,常用符号如图 1.3 所示。

算法流程图表示程序各步骤的内容以及它们之间的关系和执行的顺序,一个规范的流程图应该从唯一的椭圆代表的开始,按照有指向箭头的流程线,沿着唯一确定的路径经过若干矩形框代表的一般处理或菱形代表的逻辑判断,到达唯一的椭圆代表的结束。流程图应该足够详细,但一些常用简单的处理功能可以适当合并,以便可以顺利写出程序或检查程序的正确性。

另外,北京理工大学计算机学院副教授嵩天在他的《Python 语言程序设计基础》(第 2 版)中提到了 IPO 程序设计方法,指出每个程序都有统一的运算模式:输入数据(Input)、处理数据(Process)和输出数据(Output)。

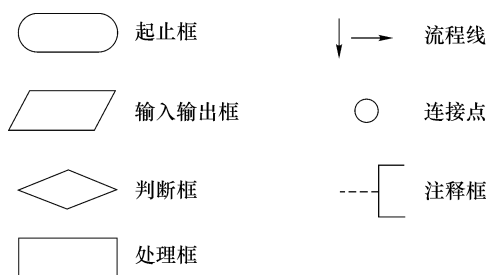


图 1.3 算法流程图常用符号

输入是用来为程序提供要处理的数据的,输入的方式有多种,包括控制台输入、文件输入、交互界面输入、随机数输入、网络输入等,本书后续章节将介绍部分输入的具体方法。

输出是供程序展示处理结果的,输出的方式也有多种,包括控制台输出、交互界面输出、文件输出等。

处理是程序对输入的数据进行计算处理产生输出结果的过程,这一部分和算法是相通的。

要写出一个好程序,一定要先做好问题分析和算法设计,然后再开始编写程序。

#### 【例 1-2】 基本信息调查程序设计。

**问题描述:**利用计算机程序调查收集学生的基本信息,包括学生的学号、姓名、计算机技能程度(陌生、一般、较好、熟练),将收集的信息整理为一段文字输出,如“学生张三的学号为 19050601,计算机操作熟练”等。

**问题分析:**要实现一个基本信息调查程序,第一,确定这个程序要给谁用;第二,了解使用者的目的是什么;第三,考虑使用者要怎么用。根据问题描述,这个程序是要收集学生的基本信息,所以使用者应为学生或者掌握学生信息的人;使用的目的是收集某位学生的学号、姓名、计算机技能程度的具体信息;当运行基本信息调查程序时,程序应逐一获取使用者提供的学号、姓名和计算机技能程度信息,然后将收集的信息整理在一起,形成一段文字描述显示给使用者。

程序设计:

##### (1) IPO 描述。

分析程序在运行过程中,需要输入什么数据、如何处理数据以及要输出什么。

##### ① Input(输入):

使用者输入学号;

使用者输入姓名;

使用者输入计算机技能程度。

##### ② Process(处理):

将使用者输入的学号、姓名、计算机技能程度信息保存并整理为一段文字。

##### ③ Output(输出):

将整理好的文字信息输出显示给使用者。

(2) 使用算法流程图,如图 1.4 所示。

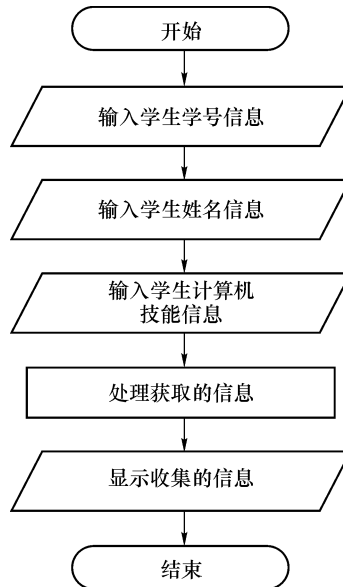


图 1.4 算法流程图

Python 程序实现:

```
# 1-2.py
# 获取信息(控制台输入)
# 提示输入“学号”,并将使用者输入的信息保存到 stuNum
stuNum = input("学号:")
stuName = input("姓名:")
stuCom = input("计算机技能(陌生、一般、较好、熟练):")

# 处理信息,将学号、姓名以及计算机技能按要求拼接为一段文字描述
info = "学生" + stuName + "的学号为" + stuNum + ",计算机操作" + stuCom

# 显示信息(控制台输出)
print(info)
```

程序运行结果如图 1.5 所示。

```
学号: 19050601
姓名: 张三
计算机技能(陌生、一般、较好、熟练): 熟练
学生张三的学号为19050601,计算机操作熟练
>>> |
```

图 1.5 基本信息调查程序运行结果图

在程序设计过程中,问题的分析、算法的设计尤为重要,而算法流程图和 IPO 方法只是描述算法的常用方法,实际中,可以根据程序的复杂度选择适当的表示方法,也可采用多种方法相结合,算法描述不应过于详细,也不能简而化之,过于粗鲁,每一个图例应能清晰反应一行或一段功能明确的代码,有效地为后续代码的编写提供参考。

## 1.2 Python 简介

Python 语言是一种开源的解释型高级程序设计语言,支持面向对象,简单易学、易用、易维护、可扩展,并且能够与多种程序设计语言完美融合。

### 1.2.1 Python 的发展

Python 是由荷兰人 Guido van Rossum(吉多·范罗苏姆)在 1989 年圣诞节期间,为打发无聊时间,而开发的一个新的脚本解释语言。

Python 继承自 ABC 语言,基于 C 语言开发。Python 的第一个公开发行人版发行于 1991 年,随后于 2002 年发布了 2.0 版本,自此 Python 转变为完全开源的开发方式,不同领域的开发者将不同领域的要点融入 Python,从而使 Python 获得了高速的发展。2008 年 Python 发布了 3.0 版本,由于 Python3 与 Python2 不兼容,导致 Python 的发展在 2008 年到 2015 年期间遇到瓶颈。但今天几乎所有 Python 主流的和最重要的库都可以运行在 Python3 上了,国际上重要的 Python 程序员也都在用 Python3,而且 2018 年 3 月,Guido 宣布将于 2020 年 1 月 1 日终止支持 Python2,因此本书选择 Python3 这个版本。

根据 IEEE 的研究报告显示,Python 的排名从 2016 年开始持续上升,2017 年、2018 年和 2019 年连续高居编程语言排行榜首位,如图 1.6 所示。在我国,Python 的兴起是从

Rank	Language	Type	Score
1	Python	☉ ☐ ☉	100.0
2	Java	☉ ☐ ☐	96.3
3	C	☐ ☐ ☉	94.4
4	C++	☐ ☐ ☉	87.5
5	R	☐	81.5
6	JavaScript	☉	79.4
7	C#	☉ ☐ ☐ ☉	74.5
8	Matlab	☐	70.6
9	Swift	☐ ☐	69.1
10	Go	☉ ☐	68.0

图 1.6 IEEE 2019 年编程语言排行榜

2017 年末开始的,这主要源于人工智能、大数据和机器学习的兴起,Python 拥有大量相关的外部库,且易学易用。此外,2017 年末全国计算机等级考试确定了从 2018 年开始新增“Python 语言程序设计”科目,这推动了 Python 在我国的发展。而随着人工智能的崛起,Python 将可能长期占据编程语言排行榜榜首的位置。

### 1.2.2 Python 的特点

Python 的编程和设计的指导原则是“优雅、明确、简单”,这在“The Zen of Python”(《Python 之禅》)中被明确表述,当我们在 Python 中执行“import this”时就会看到如图 1.7 所示的这首诗。

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

图 1.7 《Python 之禅》

Python 以简单、易学、高效著称。Python 具有简单的说明文档,初学者很容易上手,编程人员可以更专注于解决问题而不是学习编程语言本身;Python 可以让复杂的编程任务变得高效有趣,对于一个使用 Java 需要几百行代码的任务,Python 只需要十几行代码就能够完成;Python 代码良好的可读性使阅读者很容易理解开发者所写的代码,有利于团队合作。

#### 1. 跨平台性

Python 默认的解释器是用 C 语言编写的,而各种平台都支持 C 语言的编译,也就是说,Python 解释器可以在不同的平台上运行,所以用 Python 写的程序可以不经修改移植到安装有 Python 解释器的各个平台上运行,实现跨平台。

#### 2. 可扩展性

Python 提供了丰富的 API(Application Programming Interface,应用程序编程接口)和工具,使得开发人员可以轻松使用 C 语言或者 Java 语言来编写扩展模块。然而调用扩