全国计算机等级考试

二级教程

教育部考试中心

（2019年版）

# Python 语言程序设计

高等教育出版社

# 全国计算机等级考试二级教程

## ——Python 语言程序设计

## （2019 年版）

Quanguo Jisuanji Dengji Kaoshi Erji Jiaocheng

——Python Yuyan Chengxu Sheji

嵩　天　著



高等教育出版社·北京

内容提要

本书是根据教育部考试中心制订的《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）》的要求而编写的。本书共分为 11 章，第 1 章介绍程序设计的基本方法，主要包括 Python 语言概述和开发环境配置；第 2 章和第 3 章主要介绍 Python 语言基本语法元素和基本数据类型；第 4 章介绍程序的控制结构；第 5 章介绍函数和代码复用；第 6 章介绍组合数据类型；第 7 章介绍文件和数据格式化；第 8 章至第 11 章介绍 Python 的计算生态、标准库和第三方库。

本书立足自学，在内容上力求新颖，选材采用一批实用有趣的例子，以增加学习的乐趣。本教程配套自学电子教案 ppt 一套，以及全部行文代码和程序资源，可以扫码知识导图上的二维码访问这些资源，这些资源只对第一个防伪码绑定手机的读者免费使用，其他情况需付费使用。防伪码只可以绑定一次。

本教程不仅可以作为全国计算机等级考试教材，也可以作为普通高等院校及各类学校或机构的教学用书，更是一本程序设计爱好者的自学参考用书。

**图书在版编目（CIP）数据**

全国计算机等级考试二级教程. Python 语言程序设计：2019 年版 / 教育部考试中心编. -- 北京：高等教育出版社，2018.11

ISBN 978-7-04-050761-4

Ⅰ.①全… Ⅱ.①教… Ⅲ.①电子计算机-水平考试-教材②软件工具-程序设计-水平考试-教材 Ⅳ.①TP3

中国版本图书馆 CIP 数据核字（2018）第 249379 号

| | | | |
|---|---|---|---|
| 策划编辑 何新权 | 责任编辑 何新权 | 封面设计 杨立新 | 版式设计 马 云 |
| 插图绘制 于 博 | 责任校对 刁丽丽 | 责任印制 尤 静 | |

| | | | |
|---|---|---|---|
| 出版发行 | 高等教育出版社 | 网 址 | http://www.hep.edu.cn |
| 社 址 | 北京市西城区德外大街 4 号 | | http://www.hep.com.cn |
| 邮政编码 | 100120 | 网上订购 | http://www.hepmall.com.cn |
| 印 刷 | 北京市大天乐投资管理有限公司 | | http://www.hepmall.com |
| 开 本 | 787mm×1092mm 1/16 | | http://www.hepmall.cn |
| 印 张 | 15.25 | | |
| 字 数 | 370 千字 | 版 次 | 2018 年 11 月第 1 版 |
| 购书热线 | 010-58581118 | 印 次 | 2018 年 11 月第 1 次印刷 |
| 咨询电话 | 400-810-0598 | 定 价 | 42.00 元 |



物 料 号 50761-00

# 积极发展全国计算机等级考试 为培养计算机应用专门人才　促进信息产业发展作出贡献

## （序）

中国科协副主席　中国系统仿真学会理事长
第五届全国计算机等级考试委员会主任委员
赵沁平

当今，人类正在步入一个以智力资源的占有和配置，知识生产、分配和使用为最重要因素的知识经济时代，也就是小平同志提出的“科学技术是第一生产力”的时代。世界各国的竞争已成为以经济为基础、以科技（特别是高科技）为先导的综合国力的竞争。在高科技中，信息科学技术是知识高度密集、学科高度综合、具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域，迅速改变着人们的工作、生活和社会的结构，是当代发展知识经济的支柱之一。

在信息科学技术中，计算机硬件及通信设施是载体，计算机软件是核心。软件是人类知识的固化，是知识经济的基本表征，软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代，软件是信息化的核心，国民经济和国防建设、社会发展、人民生活都离不开软件，软件无处不在。软件产业是增长快速的朝阳产业，是具有高附加值、高投入高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程，促进从注重量的增长向注重质的提高方向发展。软件产业是关系到国家经济安全和文化安全，体现国家综合实力，决定21世纪国际竞争地位的战略性产业。

为了适应知识经济发展的需要，大力促进信息产业的发展，需要在全民中普及计算机的基本知识，培养一批又一批能熟练运用计算机和软件技术的各行各业的应用型人才。

1994年，国家教委（现教育部）推出了全国计算机等级考试，这是一种专门评价应试人员对计算机软硬件实际掌握能力的考试。它不限制报考人员的学历和年龄，从而为培养各行业计算机应用人才开辟了一条广阔的道路。

1994年是推出全国计算机等级考试的第一年，当年参加考试的有1万余人，2017年报考人数已达620万人。截至2017年年底，全国计算机等级考试共开考50次，考生人数累计达7 665万人，有2 885万人获得了各级计算机等级证书。

事实说明，鼓励社会各阶层人士通过各种途径掌握计算机应用技术，并通过等级考试对他们的能力予以科学、公正、权威性的认证，是一种比较好的、有效的计算机应用人才培养途径，符合我国的具体国情。等级考试同时也为用人部门录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看，不论是管理人员还是应试人员，对该项考试的内容和

形式都给予了充分肯定。

计算机技术日新月异。全国计算机等级考试大纲顺应技术发展和社会需求的变化,从2010年开始对新版考试大纲进行调研和修订,在考试体系、考试内容、考试形式等方面都做了较大调整,希望等级考试更能反映当前计算机技术的应用实际,使培养计算机应用人才的工作更健康地向前发展。

全国计算机等级考试取得了良好的效果,这有赖于各有关单位专家在等级考试的大纲编写、试题设计、阅卷评分及效果分析等多项工作中付出的大量心血和辛勤劳动,他们为这项工作的开展作出了重要的贡献。我们在此向他们表示衷心的感谢!

我们相信,在21世纪知识经济和加快发展信息产业的形势下,在教育部考试中心的精心组织领导下,在全国各有关专家的大力配合下,全国计算机等级考试一定会以“激励引导成才,科学评价用才,服务社会选材”为目标,服务考生和社会,为我国培养计算机应用专门人才的事业作出更大的贡献。

# 前　言

2018年9月，全国计算机等级考试二级Python语言程序设计科目首次考试，报考人数几倍于预期，表现出了强劲的水平测试需求。同时，Python语言科目的设立也扫清了应用型高校开设Python语言课程的障碍，再次体现了教育部考试中心在教学和学习内容方面的引领性和支撑性作用。这是一件好事！

考试不是目的，但却是有益学习的过程！

Python语言有三个重要特点：语法简洁、生态丰富、多语言集成，被称为“超级语言”。从1989年Guido van Rossum构思Python语言至今，该语言已经发展了整整30年，虽然历经波折，但在每次重大决策时刻，Python语言都以无比的勇气和卓远的见识朝向更加正确、更加先进、更加开放的方向发展。也许今天大家学习并使用Python语言开发大数据、人工智能、计算机工程等系统或软件十分自然，但从历史发展的角度来看，如今的自然在过去那个时代是多么的“不自然”，顺应时代已然不易，超越时代更加难得！

然而，历史却少有跨越式发展，平稳延续还是主流。因此，尽管我们心中的Python那么好，在编程领域，仍然是十几种语言共存共生的产业局面，仍然有不少高校尚未开设Python语言课程，仍然有大量的入门学习者不知道Python语言而对程序设计知难而退。此时，更需要鼓励创新变革的产业格局、更需要具备开拓进取精神的高校教师、更需要具有历史担当与责任感的教育管理部门，未来已来，我们需要让学生看见。

为了满足广大学生的自学和应考，教育部考试中心和高等教育出版社精心策划，围绕《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）》的要求并结合命题专家组最新的官方解读，修订后产生此书。

本教程不仅可以作为全国计算机等级考试应考用书，也可以作为高等院校及各类学校或机构的教学用书，更是一本程序设计爱好者的自学参考用书。此次改版提高了内容的广度及自学阅读的友好性，在章节设置上不仅保留了与考纲考点对应的特色，又在保持内容系统性基础上增加了实践，统筹协调了应考、教学与自学的需求。

本教程编写过程得到了教育部考试中心、高等教育出版社的大力支持和深切关怀，在此深表谢意。编写考试教程不仅需要引领方向、勇于开拓，更需要无比认真、非常细致，全书内容经过认真打磨，希望能够引领读者轻松学好Python语言，轻松考好二级科目，发现程序设计之美。

由于时间仓促、水平有限，书中仍可能存在疏漏或错误之处，敬请广大读者批评指正。

编　者

# 前　言

## （2018 年版）

考试不是目的，但却是有益学习的过程！

全国计算机等级考试是教育部考试中心组织的水平性考试，对教学或学习内容具有重要的“指挥棒”作用。很高兴地看到，在 2018 年，我国最重要的计算机类水平考试增设了 Python 语言科目。

Python 语言是一门简洁但不简单的编程语言。语法简洁、快速入门、灵活好用，任何人都可以拿来写个小程序解解闷；快速封装 C 代码、十几万生态库、大量高级语法设计，即使最专业的程序员也很难掌握其全部精髓。这样既接地气又高大上的编程语言“前无古人”，在可见的未来也是“后无来者”，更重要的是，Python 还是一门通用语言，适合任何类型的程序设计需求。

Python 语言无疑是当下最热门的编程语言。从 2013 年国内的默默无闻到 2017 年的势压群芳，Python 语言的推广和普及经过了“漫长”但“终有结果”的过程。有人说数据科学发展需要 Python，有人说人工智能火热才关注 Python，都有些道理，但从专业角度来说，Python 自有其独特且终能大成的内因，无疑，这门语言必将或已经成为计算机发展历史上具有标志性作用的重要编程语言。

伟大的精神才能产生伟大的作品。如果把 Python 语言看成作品，那么一定要向 Guido van Rossum 致敬！致敬他领导并设计了这样简洁好用的工具、致敬他坚定开源开放造福世界的精神、致敬他精深专业并勇于否定的决心。享受 Python 编程乐趣的同时，更要领会精神，以创造真正价值为导向，这样的人生才能更精彩。

为了满足广大考生学习和应考的需要，教育部考试中心组织专家经过多次研讨和反复推敲，制订并颁布了《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）》，并根据考试大纲要求组织编写了本教程。

本书在内容上力求新颖，选材尽量采用有趣的例子，每节内有一个小提示，每章开始列明学习要点，并配有知识导图辅助说明重点。教程配套自学电子教案 ppt 一套，以及全部行文代码和程序资源，可以扫码知识导图上的二维码访问这些资源，这些资源只对第一个防伪码绑定手机的读者免费使用，其他情况需付费使用。防伪码只可以绑定一次。

本教程在编写过程中得到了教育部考试中心和高等教育出版社的大力支持和深切关怀，在此深表谢意。同时特别感谢杨雅婷、李天龙两位研究生的工作，他们负责了全书所有习题答案的整理以及所有知识导图的绘制，聪明又勤奋，是祖国未来的栋梁。

由于时间仓促、水平有限，书中仍可能存在疏漏或错误之处，敬请广大读者批评指正。

编　者

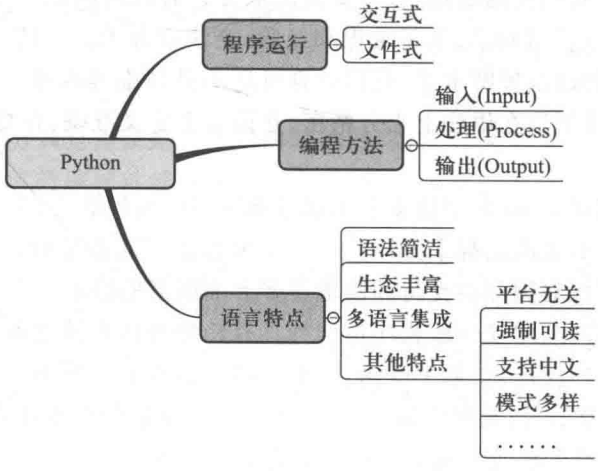
# 目　录

# 第1章 程序设计基本方法

## 学习要点

- Python 语言的特点

## 知识导图



# 1.1 程序设计语言

## 1.1.1 程序设计语言概述

程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系，它按照特定规则组织计算机指令，使计算机能够自动进行各种运算处理。程序设计语言也叫编程语言，本书两种说法都会出现并交替使用。计算机程序是按照程序设计语言规则组织起来的一组计算机指令。

程序设计语言发展经历了机器语言、汇编语言到高级语言的3个阶段。其中，机器语言和汇编语言都是直接操作计算机硬件的编程语言，只有计算机工程师在编写操作系统与硬件交互的底层程序或对程序进行反编译等情况下使用，这两类语言与具体CPU结构相关，不是当今程序设计主流方式。相比机器语言和汇编语言，高级语言是一种与计算机硬件无关、用于表达语法逻辑、更接近自然语言的一类计算机程序设计语言。

——为什么不能用自然语言，如中文，直接编写程序呢？

——因为自然语言不够精确，存在计算机无法理解的二义性。

自然语言具有不严密和模糊的缺点，需要交流双方具有较高的识别能力。例如，“我看见一个人在公园，带着望远镜。”这句话，基于常识和经验，交谈双方大多数情况下能够理解彼此表达的意思，但这需要较高的语言理解水平，现代计算机还不具备准确理解这种模糊性的完备智能。相比自然语言，程序设计语言在语法上十分精密，在语义上定义准确，在规则上十分严格，进而保证语法含义的唯一性。

从计算机诞生到应用的70余年历史上出现过600多种编程语言，至今仍然广泛使用的不超过20种，很多编程语言生命周期都十分短暂。一个编程语言能否流行，受到内因即语言设计的先进性及外因即技术时代对编程的支持和需求等多方面因素的影响。由于计算机技术发展十分迅速，需求变化也非常多样，建议一般程序员选择具有趋势性且大量使用的编程语言学习。某些编程语言简单但未必有广泛应用，如Lua语言；某些语言针对特定领域十分有用，但缺乏足够的通用性，如Matlab工具及语言；某些语言很专业，但未必有足够的产业需求及就业支撑，如Go语言。总之，学习Python语言是当下及未来几十年的正确选择！

## 1.1.2 编译和解释

高级语言根据计算机执行机制的不同可分成两类：静态语言和脚本语言，静态语言采用编译方式执行，脚本语言采用解释方式执行。例如，C语言是静态语言，Python语言是脚本语言。无论哪种执行方式，用户使用方法可以是一致的，如通过鼠标双击执行一个程序。

编译是将源代码转换成目标代码的过程。通常，源代码是高级语言代码，目标代码是机器语言代码，执行编译的计算机程序称为编译器(compiler)。如图1.1展示了程序的编译和执行过程，其中，虚线表示目标代码被计算机运行。编译器将源代码转换成目标代码，计算机可以立即或稍后运行这个目标代码。运行目标代码时，程序获得输入并产生输出。

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

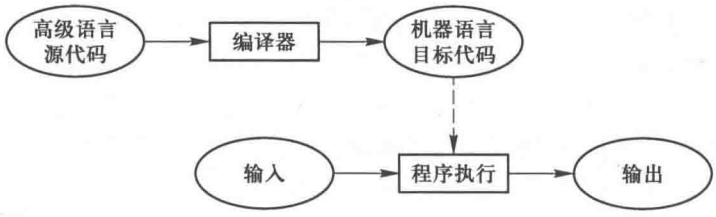



图 1.1 程序的编译和执行过程

序称为解释器(interpreter)。如图 1.2 展示了程序的解释和执行过程。其中,高级语言源代码与数据一同输入给解释器,然后输出运行结果。

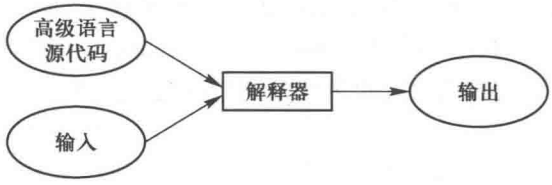



图 1.2 程序的解释和执行过程

编译和解释的区别在于编译是一次性地翻译,一旦程序被编译,不再需要编译程序或者源代码。解释则在每次程序运行时都需要解释器和源代码。这两者的区别类似于外语资料的翻译和同声传译。

简单来说,解释执行方式是逐条运行用户编写的代码,没有纵览全部代码的性能优化过程,因此执行性能略低,但它支持跨硬件或操作系统平台,对升级维护十分有利,适合非性能关键的程序运行场景。

采用编译方式执行的编程语言是静态语言,如 C 语言、Java 语言等;采用解释方式执行的编程语言是脚本语言,如 JavaScript 语言、PHP 语言等。Python 语言是一种被广泛使用的高级通用脚本编程语言,采用解释方式执行,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。这种将解释器和编译器结合的新解释器是现代脚本语言为了提升计算性能的一种有益演进。

**提示**:如何学好编程?

重要的事情说三遍:实践、实践、实践!

具体来说,从小的计算需求入手,做高质量习题,在解决问题中学习,体会编程带来的快乐。

### 1.1.3 计算机编程

——为什么要学习计算机编程?

——因为"编程"是当今众多行业技术升级和发展的主要手段,非常有用!

——因为"编程"是件很有趣的事儿,能启迪思维,还有诗和远方……

编程能够训练思维。编程体现了一种抽象交互关系、形式化执行的思维模式,称为"计算思维"。计算思维是区别于以数学为代表的逻辑思维和以物理为代表的实证思维的第三种思维模

式。编程是一个求解问题的过程，首先需要分析问题、抽象内容之间的交互关系、设计利用计算机求解问题的确定性方法，进而通过编写和调试代码解决问题，这是从抽象问题到解决问题的完整过程。计算思维的训练能够促进人类思考，增进观察力和深化对交互关系的理解。

编程能够增进认识。编写程序不单是求解计算题，它要求程序员不仅要思考解决问题的方法，更要思考如何让程序有更好的用户体验、更高的执行效率和更有趣的展示效果。不同群体、不同时代、不同文化背景对程序使用有不同理解，编程需要对时代大环境和使用群体小环境有更多认识，从细微处给出更好的程序体验，这些思考和实践将帮助程序员加深对用户行为以及社会和文化的认识。

编程能够带来乐趣。计算机编程是展示自身思想和能力的舞台，能够将程序员的所思所想变为现实。编程的开始有各种动机，或者去展示自己的青春风采，或者讽刺不文明的社会现象，或者向爱慕的对象表达情愫，所有这些想法都可以通过编写程序变成现实，并通过互联网零成本分发获得更大的影响力。这些努力会让世界增加新的颜色、让自己变得更酷，提升心理满足感和安全感。

编程能够提高效率。计算机已经成为当今社会的普通工具，掌握一定的编程技术有助于更好地利用计算机解决所面对的计算问题。例如，对于个人照片，可以通过程序读取照片元属性自动进行归类整理；对于工作数据，可以通过程序，按照特定算法进行批处理，并绘制统计图表；对于朋友圈的好文，可以通过程序实时关注随时点赞。可见，掌握一些编程技术能够提高学习、工作和生活效率。

编程带来就业机会。程序员是信息时代最重要的工作岗位之一，国内外程序员的缺口都在百万级以上规模，就业前景广阔。程序员职业往往并不需要掌握多种编程语言，精通一种就能够获得就业机会。如果读者不喜欢自己的专业或现在的工作，那就认真学习程序设计，换个更有趣的工作吧！

很多读者都有一个误区，认为编程很难学。事实上，“编程不是一件很难的事儿”，因为编写程序有一定的框架和模式，只要理解了并且稍加练习就会有很好的学习效果。学习一门编程语言，首先要对该语言的语法做到既能系统掌握，又能灵活运用。其次要学会结合计算问题设计程序结构，从程序块、功能块角度理解并设计整个程序框架。最后要掌握解决问题的实践能力，即从理解计算问题开始，设计问题的解决方法，并通过编程语言来实现。学习计算机编程的重点在于练习，不仅要多看代码，照着编写，调试运行，还要在参考代码思路的基础上独立编写，学会举一反三。为了帮助读者掌握程序设计方法，本书将大量编程的概念和语法通过有趣的实例组织起来，并展示 Python 语言的魅力和力量。希望这样的设计能够为读者在 Python 语言学习过程中带来快乐和价值。

## 1.2 Python 语言概述

——“Python 语言有 2.*x* 和 3.*x* 版本，有人建议学 2.*x* 版本，是这样吗？”

——“Python 2.*x* 已经是过去，Python 3.*x* 是这个语言的现在和未来。”

### 1.2.1 Python 语言的发展

Python 语言由 Guido van Rossum 设计并领导开发，最早的可用版本诞生于 1991 年。回顾历史，1989 年 12 月，Guido 考虑启动一个开发项目打发圣诞节假期，决定为当时正在构思的脚本语言写一个解释器，因此诞生了 Python 语言。Python 语言由大牛的“偶然”所思而诞生，但经过广大程序员近 30 年的发展和应用，Python 语言已经成为当代计算机技术发展的重要标志之一。

Python 语言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/）自由下载。

2000 年 10 月，Python 2.0 版本发布，标志着 Python 完成了自身涅槃，开启了 Python 广泛应用的新时代。2010 年，Python 2.*x* 系列发布了最后一个版本，主版本号为 2.7，用于终结 2.*x* 系列版本的发展，并且不再进行重大改进。

2008 年 12 月，Python 3.0 版本发布，这个版本解释器内部完全采用面向对象方式实现，在语法层面做了很多重大改进。这些重要修改所付出的代价是 3.*x* 系列版本代码无法向下兼容 2.*x* 系列的既有语法，因此，所有基于 Python 2.*x* 系列版本编写的代码都必须经过修改后才能被 3.*x* 系列版本解释器运行。

Python 语言经历了一个痛苦但令人期待的版本更迭过程，从 2008 年开始，用 Python 编写的几万个标准库和第三方库开始了版本升级过程，这个过程前后历时 8 年。2016 年，所有 Python 重要的标准库和第三方库都已经在 Python 3.*x* 版本下进行演进和发展。Python 语言版本升级过程宣告结束。

**提示：**如何直观判断一个 Python 程序是否为 3.*x* 版本？

最直观、最显著、最常用的判断方法是查看 print。Python 3.*x* 版本用 print() 函数替换了 Python 2.*x* 版本中的 print 语句，两者功能一样，格式不同，如下：

```
Python2.x: >>>print "The Zen of Python"
Python3.x: >>>print("The Zen of Python")
```

### 1.2.2 Python 最小程序

学习编程语言有一个惯例，即编写一个最小程序：在屏幕上打印输出“Hello World”。这个程序虽小，却是初学者了解编程语言的第一步。

Python 语言的最小程序只有一行代码，如下：

```
1 | print("Hello World")
```

上述代码中，print() 表示将括号中引号内的信息输出到屏幕上。该代码在 Python 运行环境中的执行效果如下：

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

其中，第一行的“>>>”是 Python 语言运行环境的提示符，表示可以在此符号后输入 Python 语句。第二行是 Python 语句的执行结果。