

耿肇英 编著

用 12 个流行应用程序学

Python

从入门到精通

零基础从入门到进阶

语法 + 实例由浅入深讲解

程序代码方便下载



中国出版集团有限公司



世界图书出版公司

耿肇英 编著

用 12 个流行应用程序学 Python

从入门到精通

世界图书出版公司
北京·广州·上海·西安

图书在版编目 (CIP) 数据

用12个流行应用程序学Python：从入门到精通 / 耿肇英编著. -- 北京：世界图书出版有限公司北京分公司，

2025. 1. -- ISBN 978-7-5232-1650-7

I. TP312.8

中国国家版本馆CIP数据核字第2024P28718号

-
- 书 名 用12个流行应用程序学Python：从入门到精通
YONG 12 GE LIUXING YINGYONG CHENGXU XUE PYTHON
- 编 著 耿肇英
责任编辑 张绪瑞
责任校对 王 鑫
出版发行 世界图书出版有限公司北京分公司
地 址 北京市东城区朝内大街137号
邮 编 100010
电 话 010-64038355 (发行) 64033507 (总编室)
网 址 <http://www.wpcbj.com.cn>
邮 箱 wpcbjst@vip.163.com
销 售 新华书店
印 刷 北京建宏印刷有限公司
开 本 787mm × 1092mm 1/16
印 张 24.75
字 数 620千字
版 次 2025年1月第1版
印 次 2025年1月第1次印刷
国际书号 ISBN 978-7-5232-1650-7
定 价 88.00元
-

版权所有 翻印必究

(如发现印装质量问题，请与本公司联系调换)

★学习Python的必要性

没有任何编程经验的初学者，面临的首要问题就是，首选学习哪种编程语言。无论是哪类人，强烈建议把Python作为学习的第一种编程语言。

对于非IT专业类大学生或者从业者，一般仅需要一种编程语言作为工具，用来解决本专业的实际问题，一生可能只需学习一种编程语言。Python入门容易，使用该编程语言能编写在多个操作系统运行的图形界面程序，且有大量各专业免费Python库可用，可用很少的代码，就能解决本专业的复杂问题。因此学习Python是首选。

对于IT专业类及相关专业大学生或者从业者，一般需要学习多种编程语言。把Python作为学习的第一种语言也是一个不错的选择。Python入门容易，是面向对象的编程语言，可作为掌握其它编程语言的起点。该语言也是一个强大的工具，可以开发游戏、Web应用、人工智能等多个领域的应用程序。特别是IT专业类大学生，在学习计算机基础后或同时学习并掌握Python语言，在大学第二年就可以为老师或其它公司开发一些实际的应用程序或承担项目中的部分代码，在开发中继续学习有关知识，在实际工作中学习，能更快地提高自己的工作能力，同时也积累了工作经历，为毕业求职做准备。

很多中小学生家长都想让孩子学习编程。教育部2019年发布《青少年编程能力等级》标准，将青少年编程能力考试分为图形化编程、Python编程、机器人编程和C++编程四个方向。也就是说这4种编程语言都适合青少年学习。小学低年级学生可学图形化编程语言作为启蒙，小学高年级和初高中学生再学习Python就可以了。Python语言入门容易，学习后能编写完整的图形界面应用程序和游戏，使学习的人有成就感，增加学习兴趣。大多数从小学到大学的学生可能只需学习Python就够了，只有考入IT类相关专业院校才需要学习其它编程语言。

全国青少年信息学奥林匹克联赛，简称信奥赛，由教育部和中国科协委托中国计算机学会统一组织，分为初中组和高中组，获得高中组复赛一等奖的选手有可能免试入大学或加分。由于C++目前是信奥赛唯一编程语言，因此有些家长希望孩子能获得这个免试入大学或加分的资格而首选C++，这是不理智的，获得免试入大学或加分资格的学生毕竟是极少数。C++学习难度较大，创建图形界面还需学习MFC库，还要学习其它信奥赛所需知识，所花费的精力是巨大的，

如果因此影响校内课程的学习成绩，就得不偿失了。理想的方法是首先学习Python，学得非常好，可再考虑学习C++，如C++也学得非常好，再学习信奥赛所需知识，当感觉到学信奥赛所需知识吃力，或已经影响校内课程，就应立即放弃。

★本书特色

因希望没有任何程序设计基础的不同专业各类读者，都能将本书作为学习程序设计语言的入门书，所以编者尽量选择对各专业读者都用得到的内容和例子，使不同专业读者在学完本书后，能利用自己的专业知识，选择本专业Python库，可编写本专业应用程序。绝大多数人都不会成为程序员，不需要学习过多计算机专业知识。

首先，Python语言基本语法是必学的，包括变量、数据类型、判断语句、循环语句、列表、元组、字典、函数、类和对象；要理解变量无类型，数据有类型、一切皆为对象，一切皆为对象中的引用论述，以及事件驱动概念；最后12章每章一个应用程序（application program，缩写为App），还使用了如下知识：正则表达式、随机数、二维列表反序和转置、可变和不可变数据类型、多线程、递归函数、异常处理机制、堆栈、深拷贝、用shelve保存数据、矢量图形、矢量运算等。

其次，大部分应用程序必须有一个图形界面，掌握一种创建图形界面的工具，是所有编程者必须掌握的技术。tkinter是Python自带的创建图形界面模块，可用tkinter开发一些中小型图形界面应用程序，也能为学习使用其它图形界面模块打下基础。

再次，图形图像在各个领域有广泛应用。tkinter库的Canvas类处理图形图像功能较少，而pillow库是免费开源的第三方库，提供了非常强大的图像处理功能，它能够很轻松地完成一些图像处理任务，简单易用，非常适合初学者学习。本书介绍了pillow库，并给出一些使用pillow库的例子。

最后，很多人喜欢玩游戏，也希望能编写游戏程序。为提高读者的学习兴趣，本书从12章开始，每章一个完整应用程序，共计12个，其中有5个游戏程序，1个游戏程序用pygame编写。pygame专门用来开发、设计二维电子游戏模块，是免费、开源的第三方软件包，支持多种操作系统，有良好的跨平台性。其余4个游戏是棋类游戏，典型的事件驱动，用tkinter窗体的组件完成。

★本书主要内容

本书共有23章。第1章首先说明程序设计语言概念，然后介绍Python的优缺点。第2章介绍Python安装方法，以及Python自带集成开发环境IDLE的使用。第3章到第10章是Python语言基本语法，需要在理解基础上读懂书中例子，仅仅完成书中的习题是不够的，还需在网上寻找习题练习，达到熟练掌握Python语言基本语法的目的。第11章讲解如何使用tkinter库中的组件，创建应用程序的图形界面，实现简单功能，理解事件和事件驱动的概念，以及tkinter窗体的工作机制。

从第12章到第23章，每章一个完整的应用程序，共计12个。每章开始首先说明应用程序要完成的功能和实现功能的思路。如果涉及新的知识点，用例子加以说明。为使读者更容易读懂

程序，对于主程序、每个函数和类的定义，都分别详细说明它们要完成的功能，每个函数的参数的意义，以及各个变量的意义。对于其中的难点给出详细解释。12个实例中，有11个是tkinter窗体程序，包括计算器、数字华容道、2048游戏、扫雷游戏、时钟程序、记事本程序、黑白棋游戏、画图程序、画矢量图程序、截屏程序、录屏程序；1个用pygame库编写的游戏，即投篮游戏。这些都是手机、Window系统或网上的应用程序，改用Python实现。在各章中，除了这12个程序，还有一些作为例子的应用程序，例如，使白底色透明程序、走迷宫游戏等。希望通过学习编写这些实际的应用程序，读者能掌握用Python语言编写实际应用程序的步骤和方法。

从第12章到第22章的应用程序，代码总行数逐步增加。第12章的计算器程序代码行数仅有59行；第13章的数字华容道程序，只有基本功能的3行3列数字华容道程序仅用35行；包括 3×3 、 4×4 、 5×5 完整数字华容道程序，并保证所有随机排列数字矩阵都有解，也只有79行代码。其后各章应用程序，总代码行数从100多、200多、300多，直到最后的录屏程序有500多行代码。读者在掌握Python基本语法后，可从第12章开始，按顺序学习每章的应用程序。首先要读懂程序，有些例子并不完整，要求读者自己完成这部分功能。然后看一看是否能增加功能，或者自己编写一个类似功能的程序，例如，在第12章计算器基础上，实现一个函数计算器。本书各章的应用程序也不一定只有一种实现方法，也可能不是最好的方法，读者可试一试其它实现方法。总而言之，学习编程是一个从简单到复杂的逐步提高过程，其中最重要的是自己要动手编写代码，从开始模仿别人的代码，最终达到能根据项目要求，独立完成项目，从小项目开始，到最终能完成大型项目。本书所有实例的程序代码以及因篇幅有限未能在书中出现的三个游戏——球球情侣游戏、超人游戏、小蝌蚪吃蚊虫游戏三章的文本以及实例代码都可通过以下链接（链接地址：<https://pan.baidu.com/s/1aFiw9LmnxAbQgRRPalBPgw?pwd=2025>，提取码：2025）下载获得。

在此要特别感谢北京中学的张亦弛老师，她编写了本书第四章、第五章，并逐一验证了本书的案例，为此付出了大量的心血和时间。本书是以教授施羽佳和陈彦轍同学学习Python语言时的教案为基础编写的，因此还要感谢两位同学为本书编写所提供的灵感以及为本书所做的一切工作。

书中的缺点和不足之处在所难免，敬请读者批评指正。

编者

2024年8月

第1章 Python基础 / 001

- 1.1 计算机程序设计语言 001
- 1.2 机器语言和汇编语言 001
- 1.3 高级语言及编译系统和解释系统 001
- 1.4 Python语言的优缺点 002
- 1.5 Python语言的2.x和3.x版本 003

第2章 安装Python及IDLE的使用 / 004

- 2.1 在Windows系统安装Python 004
- 2.2 Python自带IDLE使用 005
- 2.3 离开IDLE运行Python程序 006
- 2.4 Python其它的集成开发环境介绍 006

第3章 变量、字符串、数字和布尔数 / 007

- 3.1 变量 007
- 3.2 注释 010
- 3.3 垃圾自动回收器 010
- 3.4 字符串和print函数 011
- 3.5 Python提供的处理字符串函数 012
- 3.6 整数和浮点数 014

3.7 布尔(bool)类型	014
3.8 类型转换	015

第4章 input()函数和if语句 / 016



4.1 input()函数	016
4.2 最简单的if语句	016
4.3 if else语句	017
4.4 if elif else语句	018
4.5 if语句中使用运算符and、or或not	018
4.6 Python提供的其它处理字符串函数	019

第5章 while循环语句 / 021



5.1 while 循环语句格式	021
5.2 退出while 循环的方法	021
5.3 使用break语句退出循环	022
5.4 用continue语句进入下次循环	022
5.5 检查字符串能否转换为整数	023
5.6 while语句的嵌套	023

第6章 for循环语句 / 025



6.1 for循环语句的结构	025
6.2 函数range()和pass语句	025
6.3 for语句嵌套及使用break和continue	026

第7章 列表和元组 / 028



7.1 为什么要使用列表	028
7.2 创建和访问列表	028
7.3 用list()和range()创建列表	029
7.4 列表切片和备份列表	030

7.5	为列表增加元素及合并列表.....	031
7.6	删除列表中元素.....	032
7.7	修改列表指定元素的值.....	034
7.8	列表排序、反序、长度、最大和最小值.....	034
7.9	查找遍历列表元素及判断某值是否在列表中.....	036
7.10	注册和检查输入能否转换为浮点数例子.....	037
7.11	二维列表.....	039
7.12	列表数据存储方式.....	040
7.13	元组.....	040

第8章 字典 / 042



8.1	为什么要使用字典.....	042
8.2	创建字典及函数dict()和fromkeys().....	042
8.3	得到字典键对应的值.....	043
8.4	增加和修改键值对.....	044
8.5	删除键值对.....	045
8.6	用函数items()、keys()和values()遍历字典.....	045
8.7	Python的迭代器.....	046
8.8	copy()方法.....	048
8.9	使用字典格式化字符串.....	048
8.10	嵌套.....	049

第9章 函数 / 051



9.1	函数的参数、形参、实参和位置实参.....	051
9.2	关键字实参.....	052
9.3	形参默认值.....	052
9.4	局部变量、全局变量和函数返回值.....	052
9.5	可变和不可变数据类型.....	053
9.6	函数参数的值传递和引用传递.....	055
9.7	传递数量不定的实参.....	057
9.8	lambda表达式（匿名函数）.....	057

第10章 类 / 059 | 🔍

- 10.1 类和对象基本概念..... 059
- 10.2 函数__init__()和创建类对象..... 061
- 10.3 类变量、实例变量和局部变量..... 063
- 10.4 实例方法、静态方法和类方法..... 065
- 10.5 访问控制和property()函数..... 067
- 10.6 继承..... 069
- 10.7 将类的对象作为其它类的属性..... 071
- 10.8 多态性..... 073
- 10.9 以上各章练习题..... 074

第11章 用tkinter创建图形用户界面 / 075 | 🔍

- 11.1 图形用户界面、tkinter库和组件..... 075
- 11.2 修改主窗体及组件属性..... 076
- 11.3 import语句..... 078
- 11.4 Button按钮组件和事件处理函数..... 078
- 11.5 组件CheckButton和RadioButton..... 079
- 11.6 事件驱动和mainloop()方法..... 083
- 11.7 组件在窗体的布局..... 084
- 11.8 pack布局..... 084
- 11.9 grid布局和entry组件..... 086
- 11.10 place布局和Python坐标系统..... 089
- 11.11 Frame和PanedWindow组件..... 092
- 11.12 tk模块和Notebook组件..... 093
- 11.13 tkinter Tix模块..... 097
- 11.14 练习题..... 097

第12章 计算器 / 098 | 🔍

- 12.1 计算器的功能及设计思路..... 098
- 12.2 事件绑定..... 099
- 12.3 为单击按钮调用的函数增加参数的方法..... 101

12.4	实现计算器按钮矩阵.....	104
12.5	内置函数eval().....	105
12.6	用多分割符分割字符串.....	105
12.7	单击按钮共用的事件处理函数.....	106
12.8	IDLE的调试功能.....	107

第13章 数字华容道 / 110



13.1	游戏规则和设计思路.....	110
13.2	生成随机数.....	111
13.3	创建按钮矩阵.....	112
13.4	单击按钮调用的函数.....	113
13.5	用自定义类实现数字华容道游戏.....	114
13.6	增加选择游戏难度、重玩和记录游戏步数功能.....	115
13.7	判断数字华容道随机排列数字矩阵是否有解.....	118
13.8	实现数字华容道游戏只出现有解矩阵.....	119

第14章 2048游戏 / 124



14.1	游戏规则.....	122
14.2	游戏初始界面和程序框架.....	123
14.3	二维列表以及和初始化有关的函数.....	124
14.4	列表的反序和转置.....	126
14.5	切片赋值语句.....	128
14.6	矩阵4行左右移动和4列上下移动.....	129
14.7	判断游戏是否结束.....	132
14.8	玩2048游戏的一些技巧.....	132

第15章 扫雷游戏 / 133



15.1	游戏规则.....	133
15.2	程序设计思路.....	134
15.3	创建方块类.....	135
15.4	多线程.....	137

15.5	格式化字符串常量(f-string)	139
15.6	messagebox模块	140
15.7	扫雷程序初始界面和程序框架	141
15.8	单击重玩按钮要实现的功能	145
15.9	判断扫雷是否成功	146
15.10	在Label和按钮上显示图形	146
15.11	右击方块要实现的功能	148
15.12	左击到雷显示所有雷	148
15.13	用递归函数计算无雷块相邻有雷块数	149
15.14	双击触发单击和双击事件问题	151
15.15	左击方块要实现的功能	152
15.16	双击方块要实现的功能	153

第16章 秒表、定时器、闹钟和时钟 / 155



16.1	创建和导入模块	155
16.2	自定义事件	156
16.3	Notebook使时钟4功能在同一窗体	157
16.4	鼠标右击弹出菜单	158
16.5	主界面程序	160
16.6	秒表程序	162
16.7	定时器程序	164
16.8	得到日期和时间	167
16.9	世界时钟程序	169
16.10	自定义模式和非模式对话框	172
16.11	用对话框选定响铃时间和周几响铃	174
16.12	闹钟程序	177
16.13	完整主程序	181
16.14	时钟程序的组织结构	182

第17章 记事本程序 / 184



17.1	菜单组件Menu	184
17.2	菜单按钮Menubutton	186
17.3	Scrollbar组件	186

17.4	记事本程序主界面和程序框架	189
17.5	组件Text	192
17.6	查找和替换功能	197
17.7	组件Listbox和Combobox	203
17.8	颜色对话框	204
17.9	修改字体和缩放	205
17.10	用bing搜索选定字符串	207
17.11	OS模块	208
17.12	文件对话框	209
17.13	存取Text组件文本内容	210
17.14	Python异常处理机制	212
17.15	with as语句	213
17.16	实现某些功能的思路	214

第18章 用Canvas实现黑白棋 / 216



18.1	黑白棋游戏规则	216
18.2	tkinter组件Canvas	217
18.3	实现黑白棋思路	220
18.4	游戏初始界面和程序框架	222
18.5	玩家放黑子有关程序	225
18.6	deepcopy()深拷贝函数的使用	229
18.7	计算机放白子有关程序	231
18.8	黑白棋游戏程序完整程序	233

第19章 画图程序 / 234



19.1	画图程序实现的功能及思路	234
19.2	PIL图像处理库安装	235
19.3	图形文件类型及透明颜色	235
19.4	Image、ImageTk和ImageDraw类	236
19.5	在主窗体用Canvas显示PIL图形	239
19.6	使图像白底色透明程序	239
19.7	为canvas加滚动条	242
19.8	画图程序主界面和框架	243

19.9	修改线颜色和填充色.....	248
19.10	用simpledialog对话框修改线宽.....	249
19.11	修改图片尺寸.....	250
19.12	旋转和翻转图片.....	252
19.13	拖动鼠标画图形.....	253
19.14	拖动鼠标画多边形或图像.....	255
19.15	剪贴板功能.....	258
19.16	放大和缩小图形后查看.....	264
19.17	文件读写.....	265
19.18	Win 10风格工具栏.....	266
19.19	实现撤消和重做的思路.....	268
19.20	其它没有实现的功能.....	270

第20章 画矢量图程序 / 272



20.1	画程序流程图程序功能.....	272
20.2	函数find_overlapping和标签"all"和"current".....	273
20.3	组合图形.....	275
20.4	实现Win 10风格工具条第2种方法.....	278
20.5	程序主界面和框架.....	279
20.6	拖动画图形及编辑文本对象.....	282
20.7	修改鼠标光标形状.....	287
20.8	根据鼠标下方选中标记改变光标形状.....	288
20.9	增加、删除选中标记及删除选中图形.....	292
20.10	拖动图形和拖动图形边界改变图形尺寸.....	294
20.11	放大和缩小矢量图形.....	296
20.12	用shelve将数据保存为可读写文件.....	297
20.13	用shelve存取矢量图.....	299
20.14	实现画矢量图程序的文件功能.....	303
20.15	将矢量图保存为位图格式文件.....	304
20.16	一些没有实现的功能.....	305

第21章 截屏程序 / 306



21.1	PIL库ImageGrab.grab函数.....	306
------	---------------------------	-----

21.2	实现截屏程序的思路.....	306
21.3	程序初始界面和程序框架.....	308
21.4	无标题栏的透明窗体.....	310
21.5	实现定位截屏功能.....	311
21.6	实现截全屏功能.....	312
21.7	保存所截图像.....	313

第22章 录屏生成动图程序 / 314



22.1	Spinbox和Scale组件.....	314
22.2	gif格式动图概念.....	315
22.3	ImageGrab.grab方法每秒截屏数.....	317
22.4	如何获得动画的位置.....	318
22.5	显示和编辑图像的数据结构.....	320
22.6	录屏程序初始界面和框架.....	321
22.7	录屏.....	326
22.8	显示录屏图像.....	329
22.9	删除图像.....	331
22.10	撤消和重做功能.....	334
22.11	修改3个参数.....	335
22.12	播放录屏图像.....	336
22.13	检测fps值.....	338
22.14	文件菜单.....	338

第23章 投篮游戏 / 341



23.1	安装pygame.....	341
23.2	pygame游戏基本框架.....	341
23.3	pygame.Surface类和blit()方法.....	343
23.4	用Surface.blit()方法从多帧图片取出单帧.....	347
23.5	实现投篮手运球和投篮动作.....	349
23.6	投手将球投向篮筐.....	351
23.7	使用pygame.math.Vector2.....	354
23.8	用向量旋转角色.....	358
23.9	Guard类实现防守者功能.....	361

23.10	用Rect类检测碰撞.....	362
23.11	pygame显示中文.....	363
23.12	完整主程序.....	363
23.13	矩形碰撞检测缺点和圆形碰撞检测.....	365
23.14	用pygame.mask实现精准碰撞检测.....	367
23.15	用mask实现走迷宫游戏.....	370
23.16	sprite.Sprite和sprite.Group.....	371
23.17	颜色碰撞.....	373
	参考文献.....	377

第1章 Python基础

【学习导入】Python并不是一门新的编程语言，1991年就发行了第一个版本，2010年以后随着大数据和人工智能的兴起，Python又重新焕发出耀眼光芒，在历届世界编程语言排行榜中名列前茅。Python是开源免费的高级编程语言，简单易用，功能强大，非常适合作为入门编程语言。非IT专业人士也很容易掌握Python，能编写本专业的复杂程序。

1.1 计算机程序设计语言

这里的语言不是人们通常理解的英语、中文等语言，而是用字符根据一定规则书写的多条指令(命令)或语句，称为程序，用来指挥计算机完成指定任务。计算机发展历史中，出现过多种计算机程序设计语言，例如机器语言、汇编语言、C、Java和Python等语言。

1.2 机器语言和汇编语言

日常使用的手机、平板、台式机等都是计算机系统，CPU（中央处理器）是计算机系统最重要的部件，用来运行所有应用程序。能在CPU中直接运行的唯一程序设计语言是机器语言，用其它程序设计语言编写的程序都必须转换为机器语言，才能在CPU中运行。机器语言程序中有多个指令，CPU直接执行机器语言中的多条指令，完成指定任务。机器语言用二进制数代表一条指令，指令就是命令CPU完成某项工作，所有指令集合组成指令系统，各种计算机系统使用的CPU可能不同，不同的CPU有不同的指令系统，互不兼容。

编写二进制机器语言程序十分困难，因此产生了汇编语言。汇编语言用字符和数字表示机器语言的各项指令，其指令和机器语言的指令一一对应，用汇编语言编写的程序称为汇编语言源程序。相对于机器语言，编写汇编语言源程序要容易得多，但是汇编语言源程序并不能在CPU中运行，需用汇编程序将汇编语言源程序转换为机器语言程序，才能在CPU中运行。由于两者指令一一对应，从汇编语言源程序转换为机器语言程序，和直接用机器语言编写的程序，两者运行速度基本相同。这两种语言称为低级语言。

1.3 高级语言及编译系统和解释系统

编写汇编语言程序仍然比较困难，而且不同CPU的指令系统完全不同，因此产生了高级语