



易学 · 趣味 · 高效

轻松学

Python

编程

23 小时

语音教学视频

王春杰 编著

中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

王春杰 编著



轻松学 Python 编程

内 容 简 介

本书采用简单直白的语言、生动的比喻，循循善诱、深入浅出地介绍了Python的基本知识和有关应用方式。第1~第8章介绍了Python的基础知识，内容包括Python简介、Python的下载与安装、Python自带开发工具IDLE的使用、编程有关专业词汇和概念、Python中的基本数据类型、Python中的流程控制、Python的函数与模块；第9~第16章介绍了编程的进阶知识，内容包括面向对象编程、文件操作、数据库操作、错误与异常处理、测试与调试、GUI图形界面编程、并发编程、网络编程等；第17~第20章通过4个综合案例讲解Python的实战应用；第21章介绍了专业开发环境Pycharm的使用。

本书在编写时力求简单明了、通俗易懂，在讲解基础知识之后，都有或大或小的应用案例供读者实践练手，非常适合编程零基础的读者阅读学习，可帮助他们快速入门并进阶。

图书在版编目（CIP）数据

轻松学Python编程/王春杰编著. —北京：中国铁道出版社有限公司，2020.5

ISBN 978-7-113-26568-7

I. ①轻… II. ①王… 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024415号

书 名：轻松学Python编程

QINGSONG XUE Python BIANCHENG

作 者：王春杰

责任编辑：于先军

读者热线电话：010-63560056

责任印制：赵星辰

封面设计：MX DESIGN
STUDIO

出版发行：中国铁道出版社有限公司（100054，北京市西城区右安门西街8号）

印 刷：中国铁道出版社印刷厂

版 次：2020年5月第1版 2020年5月第1次印刷

开 本：787 mm×1 092 mm 1/16 印张：23.75 字数：558千

书 号：ISBN 978-7-113-26568-7

定 价：79.80元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社读者服务部联系调换。电话：（010）51873174

打击盗版举报电话：（010）51873659



王春杰

技术博主（微博@Python），网络作家（笔名“春天杰作”），微博签约自媒体人。拥有十余年Web开发、网络推广等互联网行业经验，熟悉Python、C、PHP等编程语言技术。曾在教育培训机构工作多年，目前创业中，主要从事网络与软件开发、自媒体运营，以及Python编程技术的推广和培训等工作。

责任编辑：于先军

合作咨询：QQ 46768089

封面设计：**MX** DESIGN STUDIO
O.1765628429



配套资源下载地址:

<http://www.m.crphdm.com/2020/0326/14234.shtml>

前言

年轻人是幸运的，一出生就有很多先进的技术和产品可以使用。然而笔者年龄稍大，所以几乎完整经历了国内互联网技术的发展变革，当然这也是一种财富，比如笔者在中学时代还要靠调制解调器拨号上网，而如今都是百兆、千兆光纤宽带上网。上大学时第一部手机还是只能打电话发短信的黑白小屏手机，而如今我们用的都是 4G、5G 智能上网手机，除了打电话发短信，还可以用手机上网、看电影、订票、订餐、直播等。社会日新月异，科技飞速发展，终于，人工智能的时代来临了。近几年，谷歌的人工智能围棋程序 AlphaGo 接连战胜了世界级围棋冠军李世石，又战胜了当今世界围棋排名第一的柯洁，开启了人工智能全面进军日常生活的大幕。

科技产品的更新迭代，本质是技术的更新迭代。笔者还记得在中学时面对一个黑白屏电脑的 DOS 界面打字的情景，那时候我们学习的是 BASIC 语言，后来大学时用的是 Windows98 系统，学习的是 C 语言。而如今，我们使用的都是功能多样化、智能化的操作系统，编程语言也经历了飞速的发展，比如 Python 已经出现近 20 年了，主版本也已经更新到 3.7 版本，功能越来越强大，应用越来越广泛，特别是在人工智能领域，Python 更是如鱼得水。比如我们前面提到的 AlphaGo，也用到了 Python 的有关技术。当然，Python 的用途还不止于人工智能，在大数据、云计算等方面也有广泛应用。而人工智能、区块链，这些名词在如今的媒体上经常见到，即使你学习的不是理工专业，做的不是技术类工作，想必对这些词汇也耳熟能详。

如今我们国家非常重视人工智能等技术的发展，在国家级的会议上也多次提到要大力发展人工智能技术、区块链技术，希望在世界互联网竞争新赛道上占据有利位置。而 Python 正是人工智能时代主流的语言，甚至在中小学，已经将 Python 列入计算机课程的范围。如果你稍加留意也会发现，如今在各类编程语言排行榜上，Python 也常常出现在第一名的位置上，市场份额也逐年提升。或许也正因为如此，你选择要学习 Python 编程。毕竟，就连五十多岁的某房地产大佬也在微博上宣称：“今天开始我学习一门新的语言 Python，我在寻找。这也是今天给自己人生的礼物。”足见 Python 的魅力。

当然，学习编程并不是一蹴而就的，尤其是对于零基础的人来说。不过好在 Python 本身就是一门门槛较低的编程语言，非常适合零基础的朋友作为入门语言，而且 Python 的功能也非常强大。不过即便如此，学习这门技术也需要有一个科学、系统的指引。本书是一本非常适合初学者学习参考的书籍。笔者将近年来对于 Python 的学习和研究进行了总结，并

回顾多年来自己对于各种新技术和其他几种编程语言的学习过程，结合自身的编程工作经验，编写了本书，并配套制作了教学视频（大家可以通过关注作者的微信公众号“春天杰作”并回复关键字“Python”获取下载地址）。本书内容不但有编程知识，而且有学习方法，再加上通俗易懂的表述，相信一定能够为你的 Python 之路助一臂之力。

限于水平和篇幅，本书仍有不足之处，敬请谅解。在这个即时互动的时代，作者也愿意通过网络为读者解答各类问题。读者朋友可以关注作者运营的 Python 自媒体微博（@Python）获取更多学习资料，或者通过作者个人微博（@春天杰作）个人微信公众号（“春天杰作”）与作者联系。愿我们成为朋友。

最后，赶紧开始吧！

作者

2020 年 4 月

目 录

第 1 章 人生苦短，我用 Python！——Python 简介

1.1 英明的决定	1
1.2 做一点准备	7
1.2.1 安装 Python 就这么简单	7
1.2.2 工欲善其事，IDLE	11
1.3 终于要开始写代码了，好嗨哟！	12
1.3.1 第一个 Python 程序：Hello world！	13
1.3.2 举一反三：随便 Hello 点啥	15

第 2 章 热身运动——了解一些基本概念

2.1 来点儿变化：变量	17
2.1.1 变量的赋值与使用	18
2.1.2 变量命名规则	20
2.1.3 Python 中的变量只是一个标签	21
2.2 互动起来：输出与输入	23
2.2.1 内置函数简介	23
2.2.2 输出 print()	24
2.2.3 输入 input()	26
2.3 无规矩不成方圆：注释与编码规则	27
2.3.1 注释	27
2.3.2 缩进与空行	31
2.3.3 参考 PEP8	32
2.4 物以类聚：Python 数据类型简介	34
2.4.1 六种基本数据类型简介	34
2.4.2 数据类型的判断	35
2.4.3 容器、序列、迭代、可迭代对象	36
2.4.4 布尔值	37

第 3 章 数字和数学并不可怕——数据类型之数字

3.1 数字虽多，不过三类	39
---------------------	----

3.1.1	整数	39
3.1.2	浮点数	41
3.1.3	复数	41
3.1.4	eval() 函数与 exec() 函数	42
3.1.5	进制与进制转换	43
3.2	小学数学都学过——运算符与表达式	44
3.2.1	算数运算符	45
3.2.2	比较运算符	46
3.2.3	赋值运算符	47
3.2.4	逻辑运算符	48
3.2.5	各种运算符的混合使用与优先级	48
3.3	再也不用怕数学了：数学函数与 math 模块	49
3.3.1	数学函数	49
3.3.2	math 模块	50
3.3.3	random 模块	53
3.3.4	案例：做一个机选彩票的程序	54

第 4 章 字符串也是一种“串儿”——数据类型之字符串

4.1	字符串是一种什么串儿？	55
4.1.1	字符串的三种形式	55
4.1.2	字符串的索引	58
4.1.3	字符串的判断	59
4.1.4	转义字符与原始字符串	60
4.2	字符串的 N 种“食用方法”	62
4.2.1	字符串的切片	62
4.2.2	字符串的运算	64
4.2.3	处理字符串的有关函数	67
4.2.4	字符串的有关方法	68
4.2.5	字符串的格式化输出	71
4.3	正则表达式	71
4.3.1	正则表达式基础	72
4.3.2	正则表达式的应用	73
4.3.3	re 模块	75
4.3.4	案例：验证手机号是否合法	77

第 5 章 让程序更听话——流程控制与 if 分支

5.1	送你一根指挥棒：流程控制简介	78
5.2	随机应变：if 分支	80
5.2.1	if 分支的几种结构	80

5.2.2	if 结构的嵌套	84
5.2.3	多个 if 与多个 elif 的区别	86
5.2.4	pass 语句	87

第 6 章 一句顶一万句——循环

6.1	初学编程我写了一万行代码，后来我学会了用循环	90
6.1.1	for 循环结构	90
6.1.2	range() 函数	92
6.1.3	break 与 continue	93
6.1.4	使用 for 循环遍历列表和字典	94
6.1.5	for 循环嵌套	94
6.2	更“聪明”的 while 循环	95
6.2.1	while 循环结构	95
6.2.2	while 循环与 else 配合使用	97
6.2.3	在 while 循环中使用 break	98
6.2.4	在 while 循环中使用 continue	98
6.2.5	小心死循环	99
6.2.6	while 循环嵌套	100
6.2.7	用 while 循环实现一个猜数字游戏	101
6.2.8	while 循环和 for 循环混合使用	102
6.2.9	while 循环与 for 循环之比较	103

第 7 章 我们也是有组织的——列表、元组、集合、字典

7.1	列表就是一列表，列表就是一列表	104
7.1.1	创建列表	104
7.1.2	列表的索引	105
7.1.3	列表的切片	105
7.1.4	列表的遍历	106
7.1.5	列表元素的修改、插入与删除	106
7.1.6	列表的复制	107
7.1.7	列表元素的判断	108
7.1.8	列表的嵌套	109
7.1.9	二维列表的遍历	110
7.1.10	列表的运算	110
7.1.11	操作列表的有关函数	111
7.1.12	列表的有关方法	112
7.1.13	列表解析式与生成器表达式	114
7.2	比列表“顽固”的元组	115
7.2.1	元组的创建	116
7.2.2	元组的索引、切片、删除、复制、遍历	116
7.2.3	元组的运算与判断	117
7.2.4	操作元组的有关函数	118

7.2.5	元组的有关方法	119
7.2.6	二维元组	120
7.3	字典，就像你用过的字典	120
7.3.1	字典的创建	121
7.3.2	字典元素的访问	121
7.3.3	字典元素的修改、添加与删除	122
7.3.4	字典的遍历	123
7.3.5	字典的嵌套与二维字典	124
7.3.6	操作字典的有关函数	125
7.3.7	字典的有关方法	126
7.3.8	深复制与浅复制	127
7.3.9	用字典实现多重分支	128
7.4	集合：搞搞关系	129
7.4.1	集合的创建	129
7.4.2	集合的遍历、判断与删除	129
7.4.3	集合的关系判断和运算	130
7.4.4	操作集合的有关函数	131
7.4.5	集合的有关方法	132

第 8 章 掌握核心科技——函数与模块

8.1	自制工具：自定义函数	135
8.1.1	函数的定义与调用	135
8.1.2	形参、实参、位置参数、关键字参数	136
8.1.3	参数的默认值	137
8.1.4	传递多个参数	138
8.1.5	传递列表或字典作为参数	139
8.1.6	函数的返回值	140
8.1.7	用函数实现一个猜数字游戏	140
8.1.8	变量的作用域	141
8.1.9	函数的嵌套	142
8.1.10	递归函数	143
8.1.11	匿名函数：lambda 表达式	145
8.1.12	高阶函数：map()、filter()	145
8.1.13	闭包	146
8.1.14	装饰器	147
8.2	工具箱：模块	149
8.2.1	将自定义函数存储为模块	149
8.2.2	模块的导入与使用	150
8.2.3	if <code>__name__</code> == " <code>__main__</code> ":	152

第 9 章 单身狗的渴望——面向对象编程

9.1	物以类聚，人以群分	154
-----	-----------------	-----

9.1.1	创建一个类	154
9.1.2	实例化一个对象	156
9.1.3	访问、修改类和对象的属性与方法	157
9.1.4	封装：私有属性和私有方法	159
9.1.5	继承：祖传秘方	161
9.1.6	多继承与多重继承	163
9.1.7	多态：各自表述	165
9.1.8	将类存储为模块	166
9.1.9	类模块的使用	167
9.2	模块、包、库，傻傻分不清楚？	167
9.2.1	包的创建与使用	167
9.2.2	模块、包、库的区别	168
9.2.3	Python 标准库	169
9.3	给各种包找个管家：pip	169
9.3.1	pip 的使用环境	169
9.3.2	包的安装与升级	170
9.3.3	包的查询与卸载	171

第 10 章 让鼠标休息一会儿——文件操作

10.1	文件读写	173
10.1.1	文件的打开、读取与关闭	173
10.1.2	文件的逐行读取与部分读取	174
10.1.3	查看指针位置与修改指针位置	176
10.1.4	写入文件内容	177
10.2	os 模块	178
10.2.1	目录操作	179
10.2.2	文件操作	180
10.2.3	Linux 系统下的权限操作	183
10.3	shutil 模块	183

第 11 章 让我们一起囤积数据吧——数据库操作

11.1	数据库	187
11.1.1	数据库简介	187
11.1.2	SQL 语句简介	189
11.1.3	DB-API 简介	191
11.2	Python 操作 SQLite 数据库	194
11.2.1	sqlite3 模块的使用	194
11.2.2	数据表的创建、改名与数据的添加	195
11.2.3	数据的查询、修改与删除	197
11.2.4	查看表信息与删除表	199

11.3	Python 操作 MySQL 数据库	202
11.3.1	下载安装 MySQL	202
11.3.2	使用 mysql.connector 模块操作 MySQL 数据库	211
第 12 章	人非圣贤，孰能无过——错误与异常处理	
12.1	错误与异常	215
12.1.1	错误与常见错误	215
12.1.2	异常类	218
12.2	异常处理	220
12.2.1	异常的获取与处理	221
12.2.2	使用 raise 语句自主抛出异常	223
12.2.3	使用自定义异常类	224
第 13 章	是骡子是马拉出来遛遛——调试与测试	
13.1	调试	226
13.1.1	简单的调试方法	226
13.1.2	IDLE 自带调试工具的使用	228
13.2	测试	233
13.2.1	单元测试与测试用例	233
13.2.2	unittest 模块的使用	234
第 14 章	程序也要面子——GUI 编程	
14.1	GUI 简介	237
14.2	Python 内置 GUI 模块：Tkinter	238
14.2.1	使用 Tkinter 模块创建一个界面程序	238
14.2.2	窗口、控件、事件、事件处理函数	239
14.2.3	控件布局方式	248
14.3	案例：图形版猜数字游戏	255
第 15 章	人多力量大——并发编程	
15.1	进程与线程	257
15.1.1	并发与并行	257
15.1.2	多进程与多线程	258
15.1.3	同步、异步、阻塞、非阻塞	259
15.2	多进程的使用	260
15.2.1	创建和调用子进程	261
15.2.2	通过继承 Process 类来创建多个进程	263
15.2.3	通过进程池 Pool 来创建多进程	263
15.2.4	进程间通信	265

15.3	多线程的使用	266
15.3.1	使用 <code>treading</code> 模块创建多线程	266
15.3.2	通过实例化 <code>Thread</code> 类来创建线程	267
15.3.3	线程池	267
15.3.4	线程间通信	268
15.3.5	线程安全	269
第 16 章	三天打鱼 天天上网——网络编程	
16.1	网络基础	270
16.1.1	网络的硬件基础	270
16.1.2	网络模型	271
16.1.3	网络协议	272
16.1.4	域名、DNS、URL	273
16.1.5	客户端与服务器	274
16.1.6	Socket 与 <code>socket</code> 模块	274
16.2	Web 入门	276
16.2.1	超文本与超链接	276
16.2.2	超文本标记语言：HTML	277
16.2.3	HTTP 请求	280
16.3	网络模块： <code>urllib</code>	281
16.3.1	<code>request</code> 模块	281
16.3.2	<code>error</code> 模块	283
16.3.3	<code>parse</code> 模块	284
16.3.4	<code>robotparser</code> 模块	285
第 17 章	实战项目一：让 Python 当你的工作助手	
17.1	批量处理图片	287
17.1.1	<code>pillow</code> 库简介	287
17.1.2	案例：批量调整图片大小	290
17.2	读写 Word 文档	293
17.2.1	<code>python-docx</code> 库简介	293
17.2.2	文档写入操作	294
17.2.3	文档合并操作	297
17.3	处理 Excel	299
17.3.1	<code>openpyxl</code> 库	299
17.3.2	操作 Excel 表格	300
第 18 章	实战项目二：这个虫子有点牛——网络爬虫开发	
18.1	网络爬虫简介	303
18.1.1	网络爬虫基本原理	303

18.1.2	网络爬虫的“规矩”——Robots 协议	304
18.2	requests 库	305
18.2.1	requests 库的常用方法	306
18.2.2	用 requests 库爬取一个简单网页	307
18.3	实战：爬取百度图片	310
第 19 章	实战项目三：你的游戏你做主——小游戏开发	
19.1	Pygame 库	318
19.1.1	Pygame 库简介	318
19.1.2	Display 模块	321
19.1.3	Surface 模块	322
19.1.4	图形对象的常用方法	326
19.2	案例：制作一个贪吃蛇游戏	327
第 20 章	实战项目四：确认过眼神，我遇见对的人——AI 人脸识别	
20.1	人工智能简介	334
20.2	开放平台与 API 接口	336
20.3	案例：一个人脸识别与打分项目——Python 看脸	340
20.3.1	Access Token	340
20.3.2	Base64 编码	341
20.3.3	json 数据格式	344
20.3.4	案例：Python 看脸	345
第 21 章	工欲善其事，换个 IDE？——PyCharm	
21.1	PyCharm	349
21.1.1	PyCharm 下载与安装	349
21.1.2	第一次使用 PyCharm	354
21.2	用 PyCharm 写一个聊天程序	363

人生苦短，我用 Python！ ——Python 简介

“来了？老弟！”——某网红大姐

欢迎进入 Python 的世界。你可能早就听说过“人生苦短，我用 Python！”这句话，因为它在编程界大概相当于广告界的“今年过节不收礼，收礼只收 xxx！”一样知名。可是为什么是 Python 呢？为什么不是其他编程语言？学完本章内容，你将知道答案。

本章我们将介绍 Python 的有关信息、发展状况和安装使用等基础知识，让你对 Python 有一个大体了解。我们还将介绍如何在电脑上轻松地安装 Python 环境，并体验一下使用 Python 的美妙！

1.1 英明的决定

既然已经读到了这里，那说明你可能早已决定要学习 Python 了，但无论如何，还是应该让你见识见识 Python 的厉害。你可能都没意识到，选择 Python 是一个非常英明的决定。

Python 是荷兰人 Guido Van Rossum 在 1989 年开发的一门编程语言。因为 Guido 非常喜欢一个名为 Monty Python 的喜剧团体，所以就给这门编程语言命名为 Python。因为在英语中“Python”这个单词还有“蟒蛇”的意思，所以 Python 这门编程语言的 logo 也是两条卡通蛇缠绕在一起的样子，如图 1-1 所示。



图 1-1

Python 在 1994 年发布了 1.0 版本，在 2000 年发布了 2.0 版本，在 2008 年发布了 3.0 版本。比较特殊的是，为了抛弃历史包袱，更好地适应未来的发展，Python 3.0 版本作出了重大修改，并没有向前兼容。Python 2.x 系列的最后一个版本是 Python 2.7，但 Python 2.7 也已经被 Python 团队宣布在 2020 年 1 月 1 日全面停止技术支持。目前主流的 Python 版本是 Python 3.x 系列，本书也将基于 Python 3.7 进行讲解。实际上，因为同属 3.X 大版本系列下，本书中几乎所有的知识点和代码也将适用于 Python 3.5、Python 3.6 以及 Python 官方后续会推出的 Python 3.8、Python 3.9 等版本。

从编程语言的分类来说，Python 是一门解释型的高级通用编程语言。对于计算机基础比较薄弱的读者来说可能不太理解这句话，其实你目前可以暂时不需要弄明白怎么才算解释型、高级、通用，这并不影响你学习 Python 编程。我们继续看下面的解释。

Python 是一门解释型的高级通用编程语言。这句话从三个角度给 Python 进行了分类（解释型、高级、通用），所谓的解释型语言是相对于编译型语言而言的，高级语言是相对于低级语言而言的，而通用语言是相对于专用语言而言的。新手和非计算机专业人员可以这样简单地来理解这几个概念：

解释型语言（例如 Python、PHP 等）是被计算机一边“阅读”一边执行的编程语言，类似同声传译。而编译型语言（例如 C、Java 等）是把你写的代码先进行预处理，再整体执行，相当于普通翻译。

高级语言（例如 Python、Java 等）比较接近人类语言，更利于实际使用，编程时用到的很多词汇跟人类语言（如英语等）中的一样。而低级语言（如汇编语言）是更接近计算机硬件的，不太容易被一般人看懂，但更利于计算机高效率地运行。

通用语言（例如 Python、Java 等）是指适用范围非常广泛的语言，可以用来做各种领域的编程工作。专用语言（如数据库语言 SQL）是专门用于处理某种特定事务的语言，作用局限于某个专业范围内。

理解了上面三个给编程语言分类的角度，你应该就能明白所谓“Python 是一门解释型的高级通用编程语言”这句话的含义了吧？

好了，上面只是给大家做一些简介，下面才是重点：选择 Python 的理由！

我们为什么会选择学某类技能中的具体某一种？就拿选择一门外语来说吧，你一定学了英语。显然我们的教育体制选择了英语，让小孩子从义务教育阶段甚至幼儿园就开始学英语。为什么不是俄语、日语、西班牙语？显然主要是考虑了两个角度：第一，这门语言是否足够流行强大；第二，这门语言是否简单易学。当我们以这两个角度来评价各类编程语言的时候就会发现，Python 拥有以下特点。

1. 热门流行，日益增长

编程语言有许多种，完全不亚于人类语言的数量，但是其使用范围、流行程度完全不同，就如同人类使用的语言一样，中文、英文的使用量就远远大于阿亚帕涅科语。