

# Python

## 程序设计与科学计算

尹永学 黄海涛 著

### ● 理论与实践相结合

由具有多年教学和数学建模竞赛指导经验的高校教师与具有丰富实战经验的资深算法工程师合力写作而成。

### ● 轻松入门Python

从Python安装到基本语法、函数，精选有价值的Python内容进行介绍。

### ● 科学计算+数学建模

从科学计算实战到数学建模应用实例都有所介绍，带你开启数学建模的大门。

### ● 丰富的在线资源

提供源代码文件和配套PPT文件。

# Python

## 程序设计与科学计算

尹永学 黄海涛 著

人民邮电出版社

北京

## 图书在版编目 (CIP) 数据

Python程序设计与科学计算 / 尹永学, 黄海涛著

— 北京: 人民邮电出版社, 2019.8

ISBN 978-7-115-51094-5

I. ①P… II. ①尹… ②黄… 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(2019)第067020号

## 内 容 提 要

本书由高校教师与算法工程师合作编写, 兼顾理论与实践, 层次脉络清晰, 循序渐进地展开各个知识点, 适合教学与自学。本书除了介绍 Python 程序设计方法与 Python 科学计算必备的工具包以外, 还给出了数学建模的实战案例(附带原始数据)。

本书既适合软件开发人员阅读, 也适合作为高等院校计算机相关专业的师生在 Python、科学计算、数学建模等方面的教材, 还可以作为读者自学 Python 的参考用书。

- 
- ◆ 著 尹永学 黄海涛  
责任编辑 张 爽  
责任印制 焦志炜
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
涿州市京南印刷厂印刷
  - ◆ 开本: 800×1000 1/16  
印张: 12.5  
字数: 238 千字  
印数: 1—2 400 册
- 2019 年 8 月第 1 版  
2019 年 8 月河北第 1 次印刷
- 

定价: 49.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

# 序

从人们意识到科学研究的重要性以来，科学计算（Scientific Computing）就在很多领域中被广泛使用，可见其重要程度。科学计算技术也逐渐成熟并且取得了许多出色的成果。科学计算，是指在科学与工程领域，通过数学建模与数值分析技术来分析和解决问题的过程。

在进行科学计算的过程中，往往会涉及众多的数学知识点，包括线性与非线性方程、最小二乘法、特征值、最优化、插值、积分、常微分方程和偏微分方程、快速傅里叶变换等。把抽象的数学问题与科学计算软件适当结合既有助于学生学习，也利于教师教学。那么选择什么软件比较好呢？

一直以来，科学计算领域的科学家们一般会使用 C 语言、Fortran 语言或商用数学软件 Matlab 等。这些程序设计语言或者软件固然有其各自的优点，但随着大数据、人工智能领域的发展和壮大，越来越多的科学家们开始纷纷采用 Python 作为工作“利器”。Python 具有胶水语言的特点、极为丰富的第三方库、简约性和动态性、开源属性，以及包含一组成熟且仍在进化的计算工具的生态系统，使得科学家们几乎能应付所有的工作需要，并大幅提高工作效率。基于这些原因，Python 极为适合在教学和科研中作为首选程序设计语言。目前 Python 与科学计算的相关人才缺口比较大，为了顺应形势需要，本书作者结合教学经验和实际工作经验写作了本书。

其实，Python 自诞生之日就与科学计算有着密不可分的联系。Python 的创造者 Guido van Rossum 拥有数学和计算机双硕士学位，Python 的第一个公开发行版本是其在荷兰阿姆斯特丹国家数学和计算机科学研究学会（CWI）工作期间完成的。

本书的定位是科学计算的入门书，从 Python 的安装到基本语法，从科学计算实战到数学建模应用实例都有所介绍。此外，本书还配有各章节涉及的源数据与 PPT，达到学有资源、教有资料，力求使读者从 Python 入门开始，直到拥有独立完成科学计算程序的能力。

# 前 言

## 编写背景

十多年前，我刚从国外留学回来时发现，身边的同事几乎没有人了解 Python。当然，那时 Python 的知名度和人气远没有现在这么高，加上我住在边陲小城，所以很少有人提及 Python，更不用说学习和使用它了。作为比较熟悉编程的计算数学方向的教师，我坚信 Python 将会有美好的前景，因此虽然学校没有开设 Python 的相关课程，但我仍极力给数学和统计专业的学生推荐这门优秀的程序设计语言。我当时已经预料到等他们毕业时，社会上对 Python 人才的需求量将会非常大，而且人才缺口也会很大。幸运的是，有些学生听从了我的推荐，并因此在毕业后找到了比较满意的工作，本书的另一位作者黄海涛就是其中之一。看到学生已成长为资深算法工程师，作为教师，我感到无比欣慰和自豪。

在每年一度的全国大学生数学建模竞赛中，很多赛题都是既需要数值计算，又需要统计分析的。参加比赛的学生们一般会用 Matlab、SAS、Lingo 和 C 等工具的组合，如果学生能够掌握 Python，那么就可以用它方便地解决同一种问题。Python 具有易学易用的特点，可以大大缩短学习和比赛的时间。但是这么多年来，我始终苦于没有 Python 相关的课程和合适的教材，无法把这么优秀的程序设计语言讲授给更多的学生，让他们从中受益。

随着 Python 在大数据分析、人工智能等热门领域开始发挥重要的作用，各高校也开始积极地开设相关课程。针对课程和教材需求，本书由具有多年教学和数学建模竞赛指导经验的教师与企业中具有丰富实战经验的资深算法工程师合著，力求做到理论和实践相结合。考虑到有限的学时，根据案例和实战需要，本书并没有罗列 Python 所涉及的方方面面的知识，而是进行了适当的简化，同时结合案例给出了详细的代码实现过程和运行结果。这样既能使读者满足一般的科学计算要求，又不会因内容篇幅过长而增加学习难度。

本书适合大学本科二年级以上，欲在数学建模竞赛中展示实力，并为今后深造和就业寻找易学易用工具的数学专业、统计专业、计算机专业及相关理工类专业本科生使用，也适用于在本科阶段没有学过类似课程的低年级研究生、企业工作人员和其他编程爱好者。

希望读者能够通过学习本书而感受到 Python 的美丽与强大，并与 Python 一起成长！

## 本书主要内容

本书主要包含三部分。

第一部分为基础部分，从基本的安装开始，挑选最有价值的 Python 内容进行介绍。

第 1 章介绍 Python 的历史、语言特点，以及为何用 Python 进行高效的科学计算。

第 2 章从 Python 集成环境 Anaconda 安装入手，用趣味性的语言介绍“Hello World”的由来，并系统全面地介绍实现“Hello World”的两种形式，使读者从宏观上认识 Python，了解 Python 的运行方式。同时，介绍两种 IDE Jupyter notebook 与 PyCharm，并分别阐述两者在教学与工作方面的优劣，以达到因材施教的目的。

第 3 章首先介绍注释，在学习过程中学会记笔记是非常重要的，注释就是学习编程过程中的笔记。接着介绍 Python 的标准输入以及两种格式化输出，使读者掌握输入、注释、输出 3 个重要的环节，对编程有初步的认识。

第 4 章系统介绍常量、变量和标识符，并总结 7 种运算符，概括运算符的优先级，并通过配套的实际案例进行解读。

第 5 章介绍数、字符串、列表、元组、字典和集合，并将 Python 下的数据结构操作方法进行归纳总结，讲解相应的数据结构案例。

第 6 章对控制流 if 进行深度讨论，同时介绍三元运算，为进阶学习做足准备。

第 7 章用案例与图形相结合的方式介绍 while 和 for，并结合该章的基础案例，对 break、continue 和 pass 进行介绍。

第 8 章详解函数的定义、形参与实参，并通过配套的案例来介绍必需参数、关键字参数、默认参数和可变长参数。

第 9 章归纳总结模块的引用方法，详细介绍异常处理，并引入项目中最重要完整异常处理代码流程。

第 10 章提供源数据文件，举一反三式地介绍文件的操作，同时从项目实战的角度介绍 read、readline 和 readlines 三者之间的区别。

第二部分为科学计算，详尽阐述科学计算模块 NumPy 和 SciPy，数据分析模块 Pandas，绘图模块 Matplotlib，并配有实战案例。

第 11 章介绍科学计算库 NumPy。Python 对科学计算的支持是通过不同科学计算功能

的程序包和 API 建立的。对于科学计算的每个方面，我们都有大量的选择。NumPy 是众多模块的基础，本书从 NumPy 方法入手，选取矩阵与线性代数方向的 8 个实战案例进行介绍。

第 12 章归纳了 SciPy 方法，并从实战（求积分、矩阵行列式、矩阵的逆、求解方程组、最小二乘法拟合、图像处理）方面介绍了 SciPy 在科学计算中的应用。

第 13 章从 Pandas 的数据结构开始，着重介绍数据获取与数据处理，比如缺失值填充、缺失值替换等，总结实战项目经验，同时介绍 Pandas 下的统计函数与文件读取操作。

第 14 章利用  $\sin(x)$  函数的图像逐步介绍绘图方法的思考流程，并总结 Matplotlib 的整体知识点，为数据可视化打好基础。

第三部分介绍数学建模库 Scikit-Learn，并通过回归分析开启数学建模的大门。

第 15 章从一元回归开始介绍数学模型，并结合糖尿病预测案例介绍在人工智能场景下如何通过机器学习处理数学模型。同时给出多元线性回归案例和共线性问题的解决方案，为读者开启数学建模的大门。

### 建议和反馈

由于作者水平有限，书中难免有不当之处，欢迎各位读者提出宝贵的修改意见和建议，来函请发至作者邮箱 [yxyin@ybu.edu.cn](mailto:yxyin@ybu.edu.cn)、[hhtnan@163.com](mailto:hhtnan@163.com)，或本书编辑邮箱 [zhangshuang@ptpress.com.cn](mailto:zhangshuang@ptpress.com.cn)，我们将不胜感激。读者也可以加入作者 QQ 群（867300100）进行交流。

### 致谢

感谢延边大学的各位老师和人民邮电出版社的各位编辑在本书编写过程中所提供的大力支持！感谢提供宝贵意见的同事们！感谢提供技术支持的同学们！感恩我遇到的众多良师益友！

尹永学

2018 年 11 月

# 资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

## 配套资源

本书提供如下资源：

- 源代码文件；
- 教学 PPT 文件。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

## 提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

详细信息 写书评 提交勘误

页码:  页内位置 (行数):  勘误次数:

B I U

字数统计

提交

## 扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



## 与我们联系

我们的联系邮箱是 [contact@epubit.com.cn](mailto:contact@epubit.com.cn)。

如果您对本书有任何疑问或建议，请您发邮件给我们，请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 [www.epubit.com/selfpublish/submission](http://www.epubit.com/selfpublish/submission) 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

## 关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等等。



异步社区



微信服务号

# 目 录

第 1 章 Python 概述	1
1.1 Python 语言发展史	1
1.2 Python 语言特点	2
1.3 Python 语言主要应用领域	4
1.4 本章练习	5
第 2 章 开启 Python 之旅	6
2.1 部署 Python 环境	6
2.1.1 Anaconda 简介	6
2.1.2 在 Windows 系统中安装 Anaconda	7
2.1.3 在 Linux 系统中安装 Anaconda	8
2.2 第一个 Python 程序“Hello,World”	10
2.2.1 “Hello, World”的由来	10
2.2.2 实现“Hello,World”	11
2.3 使用 Python 的 IDE	13
2.3.1 交互式解释器——Jupyter notebook	13
2.3.2 集成开发环境——PyCharm	16
2.4 本章练习	16
第 3 章 输入与输出	17
3.1 注释	17
3.1.1 单行注释	17
3.1.2 多行注释	18
3.2 输入	18
3.3 格式化输出	19
3.3.1 %操作符	19
3.3.2 format 格式化字符串	21
3.4 本章练习	22

第 4 章 变量与运算符	23
4.1 变量	23
4.1.1 常量与变量	23
4.1.2 标识符	23
4.1.3 变量初始化	24
4.2 运算符	24
4.2.1 算术运算符	25
4.2.2 比较运算符	26
4.2.3 赋值运算符	26
4.2.4 逻辑运算符	27
4.2.5 位运算符	28
4.2.6 成员运算符	28
4.2.7 身份运算符	29
4.2.8 运算符优先级	29
4.3 本章练习	30
第 5 章 数据类型与数据结构	32
5.1 数据类型	32
5.1.1 数	32
5.1.2 字符串	35
5.2 数据结构	36
5.2.1 列表	36
5.2.2 元组	38
5.2.3 字典	38
5.2.4 集合	40
5.3 本章练习	42
第 6 章 条件结构	43
6.1 if 语句	43
6.1.1 if 形式	43
6.1.2 if-else 形式	44
6.1.3 多分支选择结构	45
6.2 进阶——if 嵌套与三元运算	46
6.2.1 if 嵌套	46
6.2.2 三元运算	47

---

6.3	本章练习	48
<b>第 7 章</b>	<b>循环结构</b>	<b>49</b>
7.1	循环语句	49
7.1.1	概述	49
7.1.2	while 循环语句	49
7.1.3	for 循环语句	51
7.1.4	循环嵌套	52
7.2	break、continue、pass	53
7.2.1	break	53
7.2.2	continue	54
7.2.3	pass	54
7.3	本章练习	55
<b>第 8 章</b>	<b>函数</b>	<b>56</b>
8.1	函数概述	56
8.2	函数的参数	57
8.2.1	形式参数和实际参数	57
8.2.2	形式参数设置	58
8.3	return 语句	62
8.4	本章练习	63
<b>第 9 章</b>	<b>模块与异常</b>	<b>64</b>
9.1	模块	64
9.1.1	概述	64
9.1.2	模块的导入	65
9.2	异常	67
9.2.1	错误	67
9.2.2	异常	67
9.2.3	异常捕获与抛出	68
9.2.4	常见异常	73
9.3	本章练习	75
<b>第 10 章</b>	<b>文件操作</b>	<b>76</b>
10.1	文件的读写	76
10.1.1	概述	76

---

10.1.2	文件读取	77
10.1.3	文件写入	78
10.2	内容获取与文件指针	79
10.2.1	read、readline、readlines	79
10.2.2	文件指针	81
10.3	本章练习	83
<b>第 11 章</b>	<b>科学计算库——NumPy</b>	<b>84</b>
11.1	NumPy 简介	84
11.1.1	初识 NumPy	84
11.1.2	NumPy 安装	85
11.1.3	NumPy 的数组属性	85
11.1.4	NumPy 的数组类型	86
11.2	NumPy 创建数组	88
11.2.1	通过列表或元组转化	88
11.2.2	数学基础——矩阵	89
11.2.3	NumPy 构建特殊数组	91
11.3	索引与切片	95
11.3.1	索引机制	95
11.3.2	切片机制	96
11.3.3	切片索引	97
11.3.4	布尔型索引	97
11.4	矩阵运算与线性代数	98
11.4.1	范数计算	99
11.4.2	求逆矩阵	100
11.4.3	求方程组的精确解	101
11.4.4	计算矩阵行列式	102
11.4.5	求解特征值与特征向量	102
11.4.6	奇异值分解	103
11.4.7	QR 分解	104
11.4.8	线性方程组的最小二乘解	104
11.5	本章练习	106
<b>第 12 章</b>	<b>科学计算库——SciPy</b>	<b>107</b>
12.1	SciPy 简介	107

12.2	SciPy 应用	108
12.3	本章练习	116
第 13 章	数据分析库——Pandas	117
13.1	Pandas 中的数据结构	117
13.1.1	数据结构	117
13.1.2	Series	118
13.1.3	DataFrame	120
13.2	数据的选取	122
13.3	数据处理	125
13.3.1	缺失值删除	126
13.3.2	缺失值填充	128
13.3.3	数据替换	130
13.3.4	标识、删除重复行	132
13.4	统计函数	134
13.5	文件读取	135
13.6	本章练习	136
第 14 章	绘图工具库——Matplotlib	137
14.1	初识 Matplotlib	137
14.1.1	从 MATLAB 认识 Matplotlib	137
14.1.2	从 $\sin(x)$ 认识 Matplotlib	137
14.2	Matplotlib 功能介绍	144
14.3	本章练习	146
第 15 章	数学建模库 Scikit-Learn——以回归为例	147
15.1	Scikit-Learn 实现一元线性回归	147
15.1.1	一元线性回归理论简介	147
15.1.2	“小”数据的一元线性回归	149
15.1.3	一元线性回归分析糖尿病病情案例	152
15.2	Scikit-Learn 实现多元线性回归	157
15.2.1	多元线性回归理论简介	157
15.2.2	多元线性回归实战	158
15.3	多重共线性问题	160
15.4	本章练习	161
附录	习题答案	162

# 第 1 章

## Python 概述

---

内容概要：

- Python 语言发展史
- Python 语言特点
- Python 语言主要应用领域

### 1.1 Python 语言发展史

每个编程语言的作者都是一个技术传奇，Python 语言的发明者 Guido van Rossum 也不例外。他是荷兰人，1982 年从阿姆斯特丹大学毕业，获得了数学和计算机两个硕士学位，他最爱做的事情就是编写代码。在那个时代，硬件资源弥足珍贵，这就需要程序员们在使用 C 语言、Fortran 语言开发时，深入理解计算机的运行机制，“榨干”计算机硬件的性能。

1989 年的圣诞节，Guido van Rossum 开始编写 Python 语言的编译器，如图 1-1 所示。Python 这个名字，来自 Guido van Rossum 所挚爱的电视剧 *Monty Python's Flying Circus*。他希望 Python 成为一门功能全面、易学易用，且可拓展的编程语言。



图1-1 Python图标

1991 年，第一款 Python 编译器诞生。它是用 C 语言实现的，并能够调用 C 语言的库文件。那时的 Python 便已经具有类、函数、异常处理，包含列表和字典在内的核心数据结

构，以及以模块为基础的拓展系统。

Python 的语法很多来自 C 语言，同时也受到 ABC 语言的强烈影响。虽然 ABC 语言的一些规范至今还饱受争议（比如强制缩进），但这些语法规范使 Python 更易读。

Python 语言不仅遵从一些语言惯例，并从一开始就特别注重可拓展性，使其可以在多个层次上拓展。在高层，可以直接引入 .py 文件；在底层，可以引用 C 语言的库。这样，程序员就可以使用 Python 快速地写出 .py 文件作为拓展模块。

2000 年 5 月，Guido 和 Python 核心开发团队转移到 BeOpen.com，组建了 BeOpen PythonLabs 团队。同年 10 月，PythonLabs 团队转向 Digital Creations。2001 年，非营利组织 Python 软件基金会（PSF）成立。

Python 目前有两个版本：Python 2.7 与 Python 3.6。在撰写本书期间，最新版的 Python 是 2018 年 10 月 20 日发布的 Python 3.7.1。本书代码基于 Python 3 开发，均可在 Python 3 下运行。

## 1.2 Python 语言特点

Python 是一款开源的编程语言，这一点很重要，它以此吸引了越来越多的使用者。使用者可以自由发布、复制、阅读 Python 源代码，也可以对它进行改动，甚至把它的一部分用于新的开源软件，这反过来又拓展了 Python 的边界，形成了良性循环。

Python 拥有丰富的库，并且可移植性非常强，配合使用 C/C++ 等语言，能胜任很多工作，如科学计算、机器学习、深度学习等。

### 1. 简洁、优雅

代码是写给人看的，所以在所有程序中，代码都应该尽可能简洁，并且语法和风格应保持前后一致。一段好的 Python 程序代码就如同一篇优雅的文章，处处能体现其简洁的语言哲学，这让使用者可以专注于解决问题而不是去搞明白语言本身。

### 2. 易学、易用

Python 语法简单、模块丰富，极易上手。

### 3. 免费、开源

Python 是 FLOSS（自由/开放源码软件）之一，Python 语言本身及其科学计算模块都

可免费使用。简单地说，你可以自由地发布、复制、阅读 Python 源代码，也可以对它进行改动，甚至把它的一部分用于新的开源软件。

### 4. 高级语言

当使用 Python 语言编写程序时，用户无须考虑诸如管理程序内存一类的底层细节。

### 5. 可移植性

Python 是开源的，因此经过改动，它可以移植到许多其他平台上。而且，在一个平台上写完 Python 程序，迁移到另一个支持 Python 的平台上运行时，输出结果几乎是一样的。Python 背后的设计原则使得它可以高度扩展，这就解释了为什么现在有那么多可以解决各种任务的高级程序库。

### 6. 解释性

使用编译性语言（如 C 或 C++）开发的程序，可以从源文件（即 C 或 C++ 语言）转换为计算机使用的语言（机器语言及二进制指令集）。这个过程通过编译器和不同的标记、选项来实现。当运行程序时，连接/转载器软件把程序从硬盘复制到内存中运行。而使用 Python 语言编写的程序不需要编译成二进制代码，直接从源代码运行程序。在计算机内部，Python 解释器把源代码转换成字节码的中间形式，再把它翻译成计算机使用的机器语言运行。

事实上，使用 Python 开发程序的用户不再需要担心如何编译程序、如何确保连接转载正确的库等，所有这一切使得开发工作变得更简单。你只需要把 Python 程序复制到另外一台计算机上，它就可以工作了。

### 7. 面向对象

Python 既支持面向过程的编程，也支持面向对象的编程。在面向过程的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在面向对象的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言（如 C++ 和 Java）相比，Python 以一种非常强大又简单的方式实现面向对象编程。

### 8. 可扩展性

Python 具有与大多数主流技术相互操作的交互能力。我们可以调用不同编程语言的函数、代码、程序包和对象，例如 Matlab、C、C++、R、Fortran 以及其他语言。还有许多方法可以实现这种交互能力，例如 Ctypes、Cython 和 SWIG 等。如果你不想公开某段关键代