

面向新工科的高等学校
应用型人才培养规划教材

Python

语言及其应用

赵广辉 主编

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

面向新工科的高等学校应用型人才培养规划教材

Python 语言及其应用

Python YUYAN JIQI YINGYONG

本书特色:

(1) 以100多个实际问题求解案例为纽带,在各知识点间建立一种有机的联系,强化各知识点间的交叉融合和知识的反复再现,在培养问题求解能力的同时逐渐掌握Python程序设计相关的基础知识与应用能力。

(2) 基于Python 3.7版本编写而成,采用新形态构建形式,提供大量应用型教学案例,适合初学Python程序设计语言的读者使用;内容覆盖全国计算机等级考试(二级)Python考试大纲,可作为考试辅导用书。

责任编辑:翟玉峰 彭立辉

封面设计:张璐

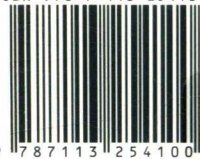
封面制作:刘颖



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

地址:北京市西城区右安门西街8号
邮编:100054
网址:<http://www.tdpress.com/51eds/>

ISBN 978-7-113-25410-0



9 787113 254100 >

定价:49.00元

面向新工科的高等学校应用型人才培养规划教材

Python 语言及其应用

赵广辉 主 编
翁 彧 副主编



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

内 容 简 介

本书融入美国斯坦福大学在 2025 计划中提出的“轴反转”教学理念，改变传统教学中以知识讲授为中心的教学组织模式，以“先能力、后知识”理念组织教学。全书以 100 多个实际问题求解案例为纽带，在各知识点间建立一种有机的联系，强化各知识点间的交叉融合和知识的反复再现，在培养问题求解能力的同时掌握 Python 程序设计基础知识与应用能力。

本书共分 10 章，内容包括 Python 语言概述、turtle 画图、基本数据类型与运算、程序流程控制、函数和代码复用、序列类型、集合与字典、异常处理、文件操作、数据分析与可视化。本书叙述清晰，案例丰富，可使学生循序渐进地学会 Python 编程技术和技巧。

本书适合作为高等学校计算机专业及其他相关专业的教材，也可作为计算机等级考试（二级）的辅导教材，以及 Python 程序设计爱好者的自学参考书。

图书在版编目（CIP）数据

Python 语言及其应用 / 赵广辉主编. —北京：中国铁道出版社，2019.7

面向新工科的高等学校应用型人才培养规划教材
ISBN 978-7-113-25410-0

I. ①P… II. ①赵… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 017720 号

书 名：Python 语言及其应用

作 者：赵广辉

策 划：秦绪好 翟玉峰

读者热线：(010) 63550836

责任编辑：翟玉峰 彭立辉

封面设计：张 璐

封面制作：刘 颖

责任校对：张玉华

责任印制：郭向伟

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）

网 址：<http://www.tdpress.com/51eds/>

印 刷：三河市航远印刷有限公司

版 次：2019 年 7 月第 1 版 2019 年 7 月第 1 次印刷

开 本：850 mm×1 168 mm 1/16 印张：17.5 字数：374 千

印 数：1～3 000 册

书 号：ISBN 978-7-113-25410-0

定 价：49.00 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，请与本社教材图书营销部联系调换。电话：(010) 63550836

打击盗版举报电话：(010) 51873659

面向新工科的高等学校应用型人才培养规划教材

编 审 委 员 会

主任：李凤霞 北京理工大学

副主任：

高林 北京联合大学

王移芝 北京交通大学

王志强 深圳大学

委员：（按姓氏笔画排序）

王若宾 北方工业大学

甘勇 郑州工程技术学院

刘丽 北京联合大学

李仲君 北京理工大学

杨小平 中国人民大学

张席 深圳技术大学

陈明 中国石油大学

赵广辉 武汉理工大学

秦绪好 中国铁道出版社

袁方 河北大学

贾宗福 哈尔滨学院

翁彧 中央民族大学

詹国华 杭州师范大学

社会的一次次重大变革、市场经济体制的不断完善使信息技术的应用范围更为广泛，也给计算机科学技术的普及应用提出了更高的要求。近年来，在我国高等教育改革力度不断加大的过程中，更加关注面向未来、兼顾当前和长期目标的高等教育。在各学科专业建设方面强调多学科交叉融合的“新工科”思维，高等工程教育融入了“成果导向”的人才培养教育理念。这种情势下，高校的计算机基础教育作为信息素养和能力培养的一个重要组成部分，面临着新一轮的机遇与挑战。

近两年来，新工科背景下产生了诸多新专业，以及相继而来的新农科、新医科、新文科等概念。一系列新专业诞生，老专业的新方案建设也在被重新定义，其中信息素养和计算机技术都无一例外地被高度关注。尤其是对新框架下的课程体系建设，新形态课程建设、教材建设、资源建设等都急迫需求。恰在此时，欣喜地看到了中国铁道出版社规划了这套“面向新工科的高等学校应用型人才培养规划教材”，将传统的计算机基础教育课程融入了体系化的思路，将目前炙手可热的“人工智能、大数据、虚拟现实、智能制造、互联网+、云计算”等新概念、新技术置入新工科框架，追求计算机技术与多学科的深度融合，以探索适应新工科背景下的专业建设和人才培养需求的教材新形态、新内容、新方法。在与出版社的多次交流中了解到，这套系列教材在组织编写思路上有很好的设计，以下几个方面值得推荐。

1. 在系列上追求教材体系化

教材体系是配合指定的课程体系构建的，而课程体系是围绕专业设置规划的。新工科背景下各专业的重塑或新建都需要在计算机公共教学过程中给予更高度融合的新的课程体系甚至是配套的教学模块。所以，本系列教材在教育部高等学校大学计算机课程教学指导委员会提出的《大学计算机基础课程教学基本要求》的基础上，向教材体系化转化，也就是说这个体系不是唯一的，而是面向多专业教学需求，可以灵活搭建不同课程体系的配套教材。本套教材，无论是在教材种类、教材形态，还是在资源配套等方面生成适用于不同专业需求的教材体系，支持不同教材体系的可持续动态增减。

2. 在内容上追求深度融合

当大数据、人工智能、虚拟现实这些新学科迅速成长起来时，如何给予计算机基础教育的多学科以支持？以往的方法是在教材里增加一章新内容，而本套教材则规划将这些内容融于不同的课程中，落地在结合专业内容的案例设计上。例如，本系列的《Python语言及其应用》一书，以近100个结合不同专业的实际问题求解案例为纽带，强化了各知识点间的交叉融合，也强化了程序设计与不同学科的融合，在宽口径的培养模式下，学生通过不断的解决问题的过程，培养了计算思维和用计算机技术解决专业问题的能力。

3. 在教学资源上同步建设

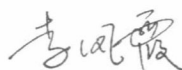
“国家教材委员会”的成立体现了教材建设的国家意志，本套教材规划起步于国家政策的高起点，除了新工科的需求牵引之外，在线教育、金课建设等大环境也要求明确，所以与教材同步的各种数字化资源建设同步规划，适时建设。从目前将要出版的其中几本教材来看，各种数字化建设都在配套开展，教学实践也在同步进行，呈现出在教材建设上的跨越式的发展态势，对教学一线教师提供了完整的教学资源，必将为新时期的人才培养目标做出可预期的贡献。

4. 教材编写与教学实践的高度统一与协同

本套教材的作者大都是教学与科研两方面的带头人，具有高学历、高职称，是具有教学研究情怀的教学一线实践者，他们设计教学过程，创新教学环境，实践教学改革，将理念、经验与结果呈现在教材中。更重要的是，在这个分享的时代，教材编写组开展了多种形式的多校协同建设，采用更大的样本做教改探索，支持研究的科学性和资源的覆盖面，必将被更多的一线教师所接受。

在当今新工科理念日益凸显其重要性的形势下，与之配合的教育模式以及相关的诸多建设都还在探索阶段，教材无疑是一个重要的落地抓手。本套教材就是计算机基础教学方面很好的实践方案，既继承了计算思维能力培养的指导思想，又融合了新工科思维，同时支持了在线开放模式。内容前瞻，体系灵活，资源丰富，是值得关注的一套好教材。

北京理工大学计算机学院教授
北京市教学名师



2019年4月

前言

传统的程序设计教学以讲授程序设计基础知识为中心，一般是先学习后应用甚至只讲授知识不涉及应用，知识学习与应用脱节严重；应用训练少，学习存在盲目性，学习动力不足。本书的内容组织反转了以“知识传授为中心”的教学模式，强调应用能力训练，使程序设计教学回归能力培养和复杂问题求解；融入有使命的学习的理念，使学习者了解学习的使命并以拥有解决复杂专业问题的能力为目标；学习与应用融为一体，成为一种基于内部驱动的使命感学习；在问题求解过程中学习成为主要的方式，在学习过程中实现个体兴趣与能力培养的最优化结合。

本书在系统讲解Python语言语法和程序设计基本思想的同时，改变了传统教学中以知识讲授为中心的教学组织模式。全书以100多个实际问题求解案例为纽带，在各知识点间建立一种有机的联系，强化各知识点间的交叉融合、反复再现，在实现问题求解的同时逐渐掌握相关的知识。全书共分10章，内容包括：Python语言概述、turtle画图、基本数据类型与运算、程序流程控制、函数和代码复用、序列类型、集合与字典、异常处理、文件操作、数据分析与可视化。在学习过程中，通过不断地解决从简单到复杂的各种问题，可赋予读者不停探索的动力，激发读者的学习兴趣和热情。

Python在大数据、人工智能、金融分析、工程问题求解等领域的核心应用都是数据的处理与可视化，紧密围绕数据处理与可视化这一核心问题的相关知识进行讲解，使读者可以具备从事机器学习和数据分析相关工作的能力。

本书基于Python 3.7版本编写而成，全书内容丰富，叙述清晰，采用新形态构建形式，并配套提供了大量应用型教学案例，适合作为高等学校计算机专业及其他相关专业的教材，以及计算机等级考试（二级）的辅导教材。

本书由武汉理工大学赵广辉任主编，中央民族大学翁或任副主编，河北大学肖胜刚、中南民族大学项巧莲、武汉理工大学董丽杰和汪朝霞、北京理工大学李仲君

参与了本书的编写工作。其中：第1~4章由赵广辉、汪朝霞、李仲君编写，第5、6章由肖胜刚、赵广辉编写，第7~9章由翁或、项巧莲编写，第10章由赵广辉、董丽杰编写。全书由赵广辉和翁或负责统稿校订，李岫、段翠苹、吴利军等参与了教学实践工作。

本书提供全套教学课件、源代码、课后习题答案与分析、常见问题及难点解析、配套实验项目、教学计划及学时分配建议。配套资源可以登录中国铁道出版社有限公司官方网站 (<http://www.tdpress.com/51eds/>) 的下载区下载或与作者联系索取，作者的微信号和QQ号为30849528，电子邮件为zhaogh@whut.edu.cn。

在本书编写过程中，我们本着科学、严谨的态度，力求精益求精，但由于水平有限，仍难免存在疏漏与不妥之处，恳请广大读者批评指正。

编者

2019年3月

目 录

第 1 章 Python 语言概述	1	1.6.3 文档注释	19
1.1 计算机语言	1	1.6.4 注释用法实例	20
1.1.1 Python语言的发展	2	小结	21
1.1.2 Python语言的特点	3	练习	21
1.1.3 Python的应用领域	3	第 2 章 turtle 画图	22
1.2 Python开发环境配置	5	2.1 turtle 库简介	23
1.2.1 解释器的安装	5	2.2 模块的导入与使用	23
1.2.2 编写Hello World程序	6	2.3 创建窗体与画布	24
1.2.3 查看帮助文档	7	2.3.1 窗体	24
1.2.4 第三方库的安装	8	2.3.2 画布	24
1.2.5 源文件打包	9	2.4 绘图实例解析	25
1.3 人机交互	10	2.4.1 绘制棋盘	25
1.3.1 输出函数print()	10	2.4.2 绘制帅府	27
1.3.2 输入函数input()	12	2.4.3 绘制兵炮标记	28
1.3.3 eval()函数	12	2.4.4 绘制棋子	29
1.3.4 简单数学计算	12	2.5 绘图函数	30
1.4 变量	13	2.5.1 画笔状态	30
1.4.1 Python关键字	13	2.5.2 颜色控制	30
1.4.2 变量的使用	14	2.5.3 颜色填充	31
1.5 编码与命名规范	15	2.5.4 辅助绘画控制	31
1.5.1 编码规范	15	2.5.5 移动与绘画函数	32
1.5.2 命名规范	18	2.5.6 返回海龟的状态函数	34
1.6 注释	18	小结	34
1.6.1 单行注释	19	练习	34
1.6.2 多行注释	19		

第3章 基本数据类型与运算	36	4.2.2 二分支结构	75
3.1 数据与数据类型	36	4.2.3 多分支结构	76
3.2 字符串类型	38	4.2.4 条件表达式	77
3.2.1 字符串的表示	39	4.2.5 pass	79
3.2.2 字符串序列操作	40	4.3 循环结构	79
3.2.3 常用字符串处理方法	44	4.3.1 for循环	80
3.2.4 字符串格式化	49	4.3.2 range()函数	80
3.2.5 转义字符	52	4.3.3 while循环	85
3.3 数值类型	53	4.3.4 break	87
3.3.1 整型	53	4.3.5 continue	90
3.3.2 浮点型	54	4.3.6 else	91
3.3.3 复数类型	55	4.4 随机数	93
3.3.4 数值类型转换	56	小结	95
3.4 迭代器类型	58	练习	95
3.5 常用运算	58	第5章 函数和代码复用	97
3.5.1 数值运算	58	5.1 函数的定义	97
3.5.2 赋值运算	62	5.2 函数的调用	99
3.5.3 比较运算	63	5.3 函数的参数传递	101
3.5.4 成员运算	64	5.3.1 位置传递	101
3.5.5 逻辑运算	65	5.3.2 关键字传递	101
3.5.6 身份运算	66	5.3.3 默认值传递	102
3.5.7 位运算	66	5.3.4 包裹传递	104
3.5.8 真值测试	67	5.3.5 解包裹传递	104
3.5.9 运算优先级	68	5.4 变量作用域	104
3.6 数学库	69	5.4.1 局部变量	105
小结	72	5.4.2 全局变量	105
练习	72	5.5 函数的返回值	107
第4章 程序流程控制	73	5.6 匿名函数	109
4.1 程序控制结构的描述	73	5.7 递归调用	111
4.2 分支结构	74	5.8 代码复用	114
4.2.1 单分支结构	74	5.9 内置函数	117

小结	119	7.2 字典	147
练习	119	7.2.1 字典的创建	147
第6章 序列类型	121	7.2.2 获取字典值	148
6.1 通用序列操作	121	7.2.3 修改字典值	150
6.1.1 索引	121	7.2.4 内置函数与方法	152
6.1.2 切片	122	7.2.5 字典排序输出	153
6.1.3 序列拼接与重复	123	7.2.6 字典综合实例	154
6.1.4 成员测试	123	小结	157
6.1.5 通用内置函数	124	练习	157
6.2 列表	124	第8章 异常处理	159
6.2.1 列表的创建	125	8.1 程序中的错误	159
6.2.2 列表的更新	125	8.2 异常	160
6.2.3 列表的删除	127	8.3 异常的处理	161
6.2.4 列表的排序	128	8.3.1 try...except子句	161
6.2.5 列表赋值与复制	131	8.3.2 单异常处理	162
6.2.6 列表推导式	131	8.3.3 多异常处理	163
6.2.7 内置函数zip() 和enumerate()	134	8.3.4 finally子句	164
6.2.8 列表嵌套	135	8.3.5 异常的应用	165
6.3 元组	137	小结	166
6.4 Range	138	练习	166
小结	139	第9章 文件操作	167
练习	139	9.1 文件的打开与关闭	167
第7章 集合与字典	141	9.1.1 文件的打开	168
7.1 集合	141	9.1.2 文件的遍历	169
7.1.1 集合的创建	142	9.1.3 文件的关闭	169
7.1.2 可变集合类型的方法	144	9.1.4 上下文管理器	170
7.1.3 成员关系	145	9.2 文件的读/写操作	170
7.1.4 集合关系	145	9.2.1 文件读取方法	171
7.1.5 集合运算	146	9.2.2 文件写入方法	174
		9.3 文件重命名与删除	175

9.4	CSV格式文件的读/写	175	10.3.7	数据统计	219
9.5	JSON文件的读/写	178	10.3.8	数据可视化	221
9.6	文件与文件夹操作	182	10.4	Seaborn的应用	222
	小结	183	10.4.1	主题	222
	练习	183	10.4.2	调色板	224
第 10 章	数据分析与可视化	184	10.4.3	数据载入	226
10.1	NumPy	184	10.4.4	单变量图	227
10.1.1	数组的属性	185	10.4.5	分布图	229
10.1.2	多维数组及其创建	185	10.4.6	箱线图	232
10.1.3	数组索引和切片	187	10.4.7	小提琴图	233
10.1.4	ufunc函数	187	10.4.8	核密度估计图	235
10.1.5	统计分析	189	10.4.9	多变量图	236
10.2	Matplotlib	191	10.4.10	回归图	239
10.2.1	绘图入门	191	10.5	词云	240
10.2.2	标注与美化	193	10.5.1	英文词云制作	241
10.2.3	绘制多子图	197	10.5.2	中文词云制作	242
10.2.4	根据数据文件绘图	198	10.6	网络爬虫	244
10.2.5	绘制饼图	203	10.6.1	网络爬虫的分类	244
10.2.6	绘制直方图	205	10.6.2	爬虫的基本原理	245
10.2.7	绘制雷达图	206	10.6.3	网络爬虫的常用技术	245
10.2.8	绘制散点图	208	10.6.4	网络爬虫开发常用 框架	250
10.2.9	绘制等值线图	209	10.6.5	爬虫的实例	252
10.2.10	绘制三维图	210		小结	255
10.2.11	曲线拟合	211		练习	256
10.3	Pandas的应用	213	附录 A	常见问题及解答	257
10.3.1	数据结构	213	附录 B	常用函数	261
10.3.2	读取数据库中数据	214	附录 C	ASCII表	265
10.3.3	读取文件中数据	216	参考文献	266
10.3.4	数据存储	217			
10.3.5	数据查看	219			
10.3.6	数据排序	219			

第1章 Python 语言概述

Life is short, you need Python! 是关于Python的一句经典的、富有情怀的语句，很多Python程序开发者都是从这句话开始学习并爱上这门语言的，使Python从众多编程语言中脱颖而出。

30年前，编程可能只是计算机行业才需要的技能，今天，编程几乎已经成为所有脑力劳动者必备的基础技能。学习简单的编程，并不比学习使用Office软件难。现在的许多编程语言（如Python）已经十分友好。学习者不需要懂得内存地址、线程等复杂的概念就可以写出一个实用性程序。

程序设计实践性极强，需要大量的编码练习才能够熟练掌握，初学者只有通过不断的实践，才能更快地学会编程！

学习目标：

- 配置Python开发环境。
- 掌握简单的人机交互。
- 了解字符串转数值。
- 了解简单的数学运算。

1.1 计算机语言

计算机程序是一组让计算机执行一系列动作的指令集。在计算机系统中，通常把显示器、主机和硬盘等看得见摸得着的部件称为硬件，它们是计算机系统的物质基础。仅有硬件而没有软件，计算机也不能工作。计算机系统必须要配备完善的软件系统且充分发挥其硬件的各种功能才能正常工作。这里的软件是指计算机运行所需的各种程序以及相关数据和文件的集合，所以人们经常说：软件=程序+文档。

总的来说，计算机语言可以分为机器语言、汇编语言、高级语言三大类。

机器语言是由0和1构成的代码，是计算机能够识别的唯一一种语言，但它非常难于记忆和识别。通常人们编程时，并不采用机器语言。汇编语言用一条指令对应实际操作过程中的一个很细微的动作，例如移动、相加等，因此汇编源程序一般比较冗长、复杂、容易

出错，而且使用汇编语言编程需要有更多的计算机专业知识。汇编语言的优点是可以直接面向硬件，生成的可执行文件小，执行速度很快。这两种语言都是面向机器的语言，与具体机器的指令系统密切相关，称为低级语言。

高级语言是以人类的自然语言和数学公式为基础的一种编程语言，基本脱离了机器的硬件系统，用人们更容易理解的方式编写程序。它将许多相关的机器指令合成为单条指令，省略了很多细节，编程者不需要有太多的专业知识就可编程。高级语言的出现使编程的难度大幅降低，同时也提高了编程的速度。

用高级语言编写的程序称为源程序，不能直接被计算机识别，必须经过编译或解释成机器语言才能被执行。

解释执行时应用程序源代码一边由相应语言的解释器翻译成机器语言，一边执行，效率比较低，不能生成独立的可执行文件，应用程序不能脱离其解释器。但这种方式比较灵活，可以动态地调整、修改应用程序。

编译是指在源程序执行之前，就将程序源代码翻译成机器语言，可以脱离其语言环境独立执行，使用比较方便、效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译才能执行，修改不方便。

高级语言并不是特指某一种具体的语言，而是包括很多编程语言，如流行的Java、C、C++、Python、C#、Visual Basic等都属于高级语言。这些语言的语法、命令格式都不相同。Java、C、C++等大部分语言都是面向程序员设计的，着重于性能和编程的灵活性，语法比较复杂。

Python与其他语言相比，是最接近自然语言的程序设计语言，关键字少、结构简单、语法清晰。它没有其他语言通常用来访问变量、定义代码块和进行模式匹配的命令式符号，代码变得更加清晰和易于阅读，使得学习者可以在相对更短的时间内掌握编程方法。此外，借助于其丰富的第三方库，可以快速地完成开发任务。

Python是一种简洁优美且设计优秀的通用编程语言，在各领域的应用几乎是没有限制的，可以完成现实中各个领域的各种任务，如科学计算、数据处理、可视化、图像处理、网站运维、自然语言处理、Web开发、机器学习、大数据、数据挖掘、人工智能等。近几年，由于大数据和人工智能领域的飞速发展，Python借助于其极其强大的科学计算和数据分析能力，得到越来越广泛的应用。

1.1.1 Python语言的发展

Python的作者Guido von Rossum是荷兰人。1982年，Guido从阿姆斯特丹大学获得了数学和计算机硕士学位。尽管拥有数学和计算机双料资质，但他总趋向于做与计算机相关的工作，并热衷于做任何和编程相关的事。那时，他接触并使用过诸如Pascal、C、Fortran等语言。这些语言的基本设计原则是让机器能更快地运行。为了增进效率，迫使程序员尽可能像计算机一样思考，以便写出更符合机器特点的程序，这使得整个编写过程需要耗费大量的时间。

1989年, Guido开始写Python语言的编译/解释器。Python来自Guido所挚爱的电视剧*Monty Pythors' Flying Circus*, 他希望这个新的称为Python的语言, 能成为一种介于C和Shell之间、功能全面、易学易用、可拓展的语言。

1991年, 第一个公开发行的Python编译器(同时也是解释器)诞生。它是用C语言实现的, 并能够调用C库(.so文件)。从一开始, Python已经具有了类(class)、函数(function)、异常处理(exception)、包括表(list)和字典(dictionary)在内的核心数据类型, 以及模块(module)为基础的拓展系统。图1.1所示为最初的Python logo。



图1.1 最初的Python logo

Python将许多机器层面上的细节隐藏起来, 交给编译器处理, 并凸显出逻辑层面的编程思想。Python程序员可以花更多的时间用于思考程序的逻辑, 而不是具体的实现细节, 这一特征吸引了广大的程序员。

2000年10月, Python 2.0正式发布, 开启了Python语言广泛应用的新时代。由于Python 2.x的巨大成功, Python语言在2007、2010和2018年三度赢得Tiobe年度编程语言奖项。

2008年12月, Python 3.0正式发布, 相对于早期版本, 3.0做了较大的升级。为了不带入过多的累赘, Python 3.0在设计时没有考虑向下兼容, 所以目前是Python 2.x和Python 3.x共存。本书将以Python 3.7.0为基础进行讲解, 书中的示例和讲解内容都是基于这个版本进行的。

1.1.2 Python语言的特点

Python语言具有以下特点:

- (1) 易于学习: Python结构清晰, 语法简洁, 学习起来更加简单。
- (2) 易于阅读: Python代码定义得更清晰。
- (3) 易于维护: Python的成功在于其源代码非常容易维护。
- (4) 一个广泛的标准库: Python的最大优势之一是具有跨平台的、丰富的库, 在UNIX、Linux、Windows和Macintosh之间具有良好的兼容性。
- (5) 易于移植: 基于其开放源代码的特性, Python可以被移植到许多平台。
- (6) 易于扩展: 关键代码或追求性能的部分代码可用C或C++完成, 在Python程序中调用。
- (7) 数据库支持: Python提供所有主流数据库的接口。
- (8) GUI编程: Python支持GUI, 可以创建和移植到许多系统调用。
- (9) 可嵌入: 可以将Python嵌入到C/C++程序, 让程序的用户获得“脚本化”的能力。

1.1.3 Python的应用领域

计算机从诞生发展到现在, 全世界有超过2 500种有文档资料的计算机语言, 但真正活跃的语言不到100种, 而最活跃的20种语言大约占据80%的市场。其中应用最广泛的Java、

C、C++、Python和C#这5种语言占据半壁江山。

Python 语言因其简单、易用、通用、严谨的特点，高居IEEE编程语言排行榜首位和Tiobe排行榜第3位，成为人工智能（AI）和大数据时代的第一开发语言，也是目前应用最广泛的编程语言之一。

在AI产业领域，顶尖科学家、机器学习专家和算法专家仅占不到5%，95%甚至更多的AI从业人员都是来自各行各业，他们掌握各自领域知识和数据资源，其主要工作是分析和处理数据。对于这些人员来说，Python 拥有非常好的计算生态，拥有丰富的数值算法和强悍的数据处理方法，拥有易学易用、高效开发等特点，加之基于Python的PyTorch、Keras、TensorFlow等众多深度学习框架的广泛应用，毫无疑问使Python成为首选工作语言。

Python 语言在人工智能领域独领风骚，在其他领域也有极好的表现：

1. 科学运算与数据可视化

从1997年开始，NASA就在大量使用Python进行各种复杂的科学运算，随着NumPy、SciPy、Matplotlib等众多程序库的开发和完善，使得Python越来越适合于进行科学计算、绘制高质量的2D和3D图像。Python是一门通用的程序设计语言，比其他脚本语言的应用范围更广泛、更灵活。

数据处理是 Python 最重要的应用领域之一，例如，从搜索数据处理到股票数据量化分析，再到火星探测数据挖掘等领域，Python都有极其广泛的应用。而Seaborn、Bokeh、Glean、Plotly、PyQtDataVisualization、Pygal、dataswim、geoplotlib、ggplot、missingno、vispy等功能强大的第三方库更为Python在数据处理与可视化领域的霸主地位奠定了坚实的基础。

2. 金融分析

在金融工程领域，Python用得最多，而且重要性逐年提高。Python语言结构清晰简单、库丰富、成熟稳定、科学计算和统计分析功能强大，编程效率远远高于C、C++和Java，尤其擅长策略回测。金融行业的很多分析程序、高频交易软件都是用Python开发。目前，Python是金融分析、量化交易领域中用得最多的语言。

3. Web开发

Python下有许多款不同的 Web 框架，Django是其中最有代表性的一个，许多成功的网站和APP都基于Django建构而成。

Flask是一个使用 Python 编写的 Web 应用框架，花很少的成本就能够开发一个简单的网站。

Tornado是一种 Web 服务器软件的开源版本，它是非阻塞式服务器，速度相当快，每秒可以处理数以千计的连接，是实时 Web 服务的一个理想框架。

4. 自动化运维

大数据时代，服务器、存储设备的数量越来越多，大数据集中趋势越来越明显，网络也变得更加复杂，用户体验和数据时效性要求更高。IT运维对实时采集和海量分析要求更高，Python以其数据处理能力强、可移植性强、开发效率高和兼容性相对其他脚本语言好等特点，几乎成为所有运维人员尤其是Linux运维人员必须掌握的程序设计语言。