



STEM 创新教育系列

青少年

一本有趣的 Python 书!

左利鑫 史卫亚 / 编著

Python 编程入门

快速入门

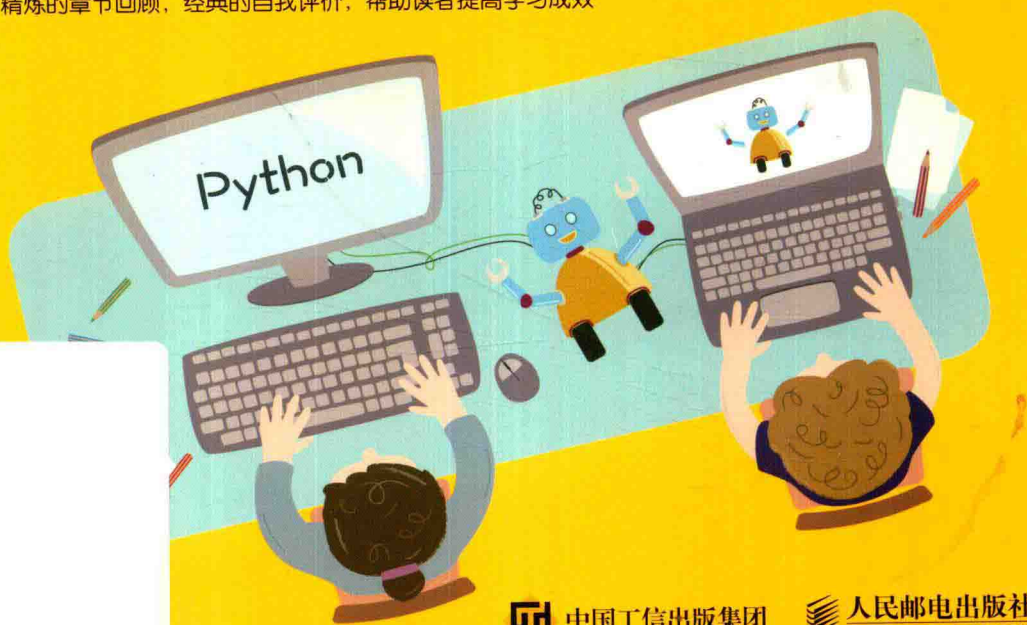
图文并茂，理论与实践相结合，引导读者快速入门

轻松上手

有趣的文字叙述，细致的流程讲解，帮助读者学习程序轻松上手

熟练应用

精炼的章节回顾，经典的自我评价，帮助读者提高学习成效



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



STEM 创新教育系列

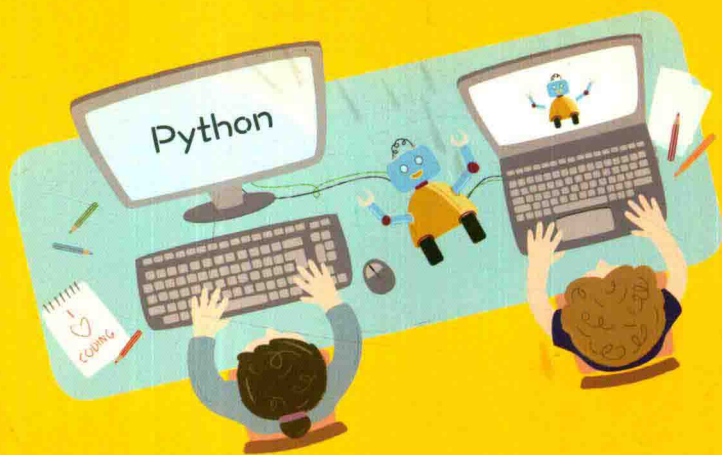
青少年

Python 编程入门

Python 可以用来做机器人，通过编程实现人工智能！

Python 学习不需要任何基础，中小學生也可以学会！

Python 应用前景广阔，一招在手，天下我有！



关注职场研究社
获取海量学习资源

封面设计：董志桢

分类建议：计算机 / 程序设计

人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-115-51014-3



9 787115 510143

ISBN 978-7-115-

定价：45.00

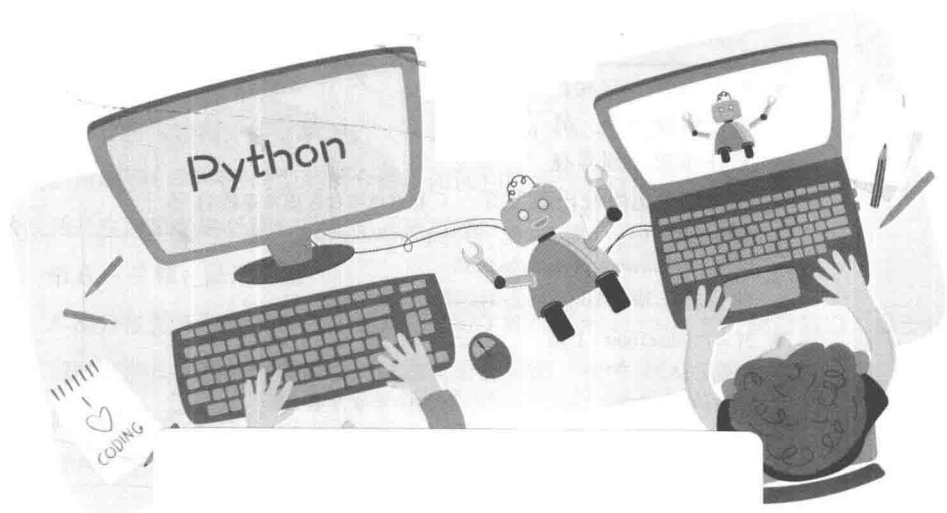


STEM 创新教育系列

青少年

左利鑫 史卫亚 / 编著

Python 编程入门



人民邮电出版社

北京

此为试读, 需要完整PDF请访问: www.ertongbook.com

图书在版编目 (C I P) 数据

青少年Python编程入门 / 左利鑫, 史卫亚编著. —
北京: 人民邮电出版社, 2019.6 (2019.9重印)
ISBN 978-7-115-51014-3

I. ①青… II. ①左… ②史… III. ①软件工具—程
序设计—青少年读物 IV. ①TP311.561-49

中国版本图书馆CIP数据核字(2019)第069093号

内 容 提 要

Python 可以用来做机器人!通过编程实现人工智能!

Python 不需要任何基础!中小學生也能学会!

Python 前景广阔!一招在手, 天下无敌!

.....

在纷繁的编程世界中, Python 毫无疑问已经成为非常适合青少年学习的语言。它入门简单, 应用广泛, 青少年既能玩得开心, 又可以为将来的深造打下基础。

本书就是专为青少年打造的 Python 入门读物。全书图文并茂, 讲解细致, 从搭建开发环境入手, 逐步引导读者掌握 Python 的基础知识、核心操作及编程技巧, 最后深入了解编程思维。

本书适合 Python 语言的零基础读者学习, 尤其适合青少年读者阅读使用。此外, 对中小学人工智能相关课程及青少年编程培训班的授课教师, 也有一定的参考作用。

◆ 编 著 左利鑫 史卫亚

责任编辑 张 翼

责任印制 周昇亮

◆ 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>

北京鑫丰华彩印有限公司印刷

◆ 开本: 700×1000 1/16

印张: 15.75

字数: 342 千字

印数: 2 501-3 500册

2019年6月第1版

2019年9月北京第2次印刷

定价: 45.00 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

青少年为什么要选择学习 Python?

与其他编程语言相比, Python 学习起来非常简单, 但功能却十分强大。从电脑上常见的应用软件, 到手机上热门的 APP, 再到近年来火热的人工智能、机器人、大数据……到处都有 Python 的身影。青少年掌握 Python 后, 可以对编程思维有深入的理解, 并为将来的深造打下坚实的基础。

本书讲了什么?

本书根据青少年的学习能力组织内容, 由浅入深介绍知识点及其应用方法, 帮助读者领略 Python 的魅力! 全书共分 3 个部分。

第 1 ~ 3 章: 基础入门

要学习 Python 语言, 首先必须搭建好开发环境, 然后在此基础上学习数据类型、表达式、运算符及流程控制语句等基础知识。青少年在学习完这部分内容之后, 可以大致了解编程的基本思维, 并利用 Python 编写语句实现简单的功能。

第 4 ~ 7 章: 知识进阶

该部分将带领青少年了解 Python 语言的核心知识, 包括字符串、序列、函数、模块及函数库等。其中的典型范例介绍了如何利用序列进行排序与求和, 以及如何通过函数快速产生随机数并处理日期和时间数据等。

第 8 ~ 9 章: 图形界面

本部分在面向对象思想的基础上, 介绍了如何利用 tkinter 套件建立窗口, 以及在窗口上添加各种控件等操作, 帮助读者开发出“看得见”的软件。

本书适合谁学习?

本书由左利鑫、史卫亚和岳福丽三位老师共同编著, 适合 Python 语言的零基础读

者学习，尤其适合青少年读者阅读使用。此外，对中小学人工智能相关课程及青少年编程培训班的授课教师，也有一定的参考作用。

编者

2019年1月

001	第1章 打造 Python 世界
002	1.1 认识 Python
002	1.1.1 Python 的版本
003	1.1.2 安装 Python 软件
003	1.2 青春行——构建 Python 环境
003	1.2.1 下载 Python 软件
005	1.2.2 安装 Python 软件
007	1.2.3 测试 Python 环境
011	1.3 逛一逛 Python 大观园——IDLE (Integrated Development Environment)
011	1.3.1 Python 的 IDE 软件
011	1.3.2 启动 IDLE 软件
012	1.3.3 Python Shell 交互模式
013	1.3.4 Edit 窗口编写程序代码
015	1.3.5 用函数 help() 获得更多帮助
017	1.4 第一个 Python 程序
017	1.4.1 开始写 Python 程序
019	1.4.2 程序的注释
020	1.4.3 语句的分行和合并
021	1.4.4 程序的输入和输出
023	1.5 新手上路
025	第2章 Python 百宝箱
026	2.1 存储数据的变量
026	2.1.1 关键字
027	2.1.2 标识符的命名规则
028	2.1.3 变量赋值

029	2.1.4 交换变量值
031	2.2 Python 的整数类型
032	2.2.1 整数
033	2.2.2 布尔类型
034	2.3 Python 的浮点数类型
034	2.3.1 Float 类型
037	2.3.2 复数类型
038	2.3.3 Decimal 类型
039	2.3.4 认识有理数
040	2.4 认识表达式
041	2.4.1 算术运算符
042	2.4.2 两数相除和相关的运算符
043	2.4.3 代数问题
044	2.4.4 math 模块进行数值运算
047	2.5 赋值运算符
048	2.6 逻辑、比较运算符
048	2.6.1 比较运算符
049	2.6.2 逻辑运算符
053	第 3 章 使用控制流
054	3.1 程序控制简介
054	3.1.1 常用的流程符号
055	3.1.2 程序子块和缩进
056	3.2 选择结构
056	3.2.1 if 语句单向选择
058	3.2.2 if/else 语句做双向选择
061	3.2.3 认识三元运算符
062	3.2.4 多重选择
066	3.3 while 循环
066	3.3.1 while 循环的特性

068	3.3.2 在 while 循环中加入计数器
071	3.4 for/in 循环
072	3.4.1 内置函数 range()
075	3.4.2 嵌套循环
077	3.5 continue 和 break 语句
077	3.5.1 break 语句
078	3.5.2 continue 语句
081	第 4 章 字符串
082	4.1 由字符说起
082	4.1.1 认识字符函数
082	4.1.2 转义字符
084	4.2 字符串的特色
084	4.2.1 建立字符串
085	4.2.2 字符串与运算符
087	4.2.3 字符串具有不变性
087	4.2.4 for/in 循环读取字符串
088	4.3 切片的概念
088	4.3.1 切片的运算
091	4.3.2 内置函数 slice()
092	4.4 处理字符串常用函数
092	4.4.1 变更字符串的大小写
093	4.4.2 字符串的搜索和替换
097	4.4.3 字符串的分割和结合
099	4.4.4 将字符串对齐
100	4.5 格式化字符串
101	4.5.1 格式运算符 %
103	4.5.2 内置函数 format()
105	4.5.3 str.format() 方法

111	第 5 章 组合不同的数据
112	5.1 认识序列类型
112	5.1.1 序列类型的特点
113	5.1.2 序列元素及操作
115	5.2 Tuple
116	5.2.1 建立 Tuple
116	5.2.2 内置函数 tuple
117	5.2.3 Index() 和 count() 方法
119	5.2.4 读取 Tuple 元素
122	5.2.5 Tuple 和 Unpacking
124	5.3 List 的基本操作
124	5.3.1 生成 List 对象
126	5.3.2 与 List 有关的方法
129	5.4 将数据排序与求和
129	5.4.1 list.sort() 方法
130	5.4.2 Tuple 元素的排序
133	5.4.3 内置函数 sum()
134	5.5 当 List 中还有 List
135	5.5.1 生成矩阵
136	5.5.2 读取二维 List
138	5.6 认识 List 生成式
139	5.6.1 为什么要有生成式?
139	5.6.2 善用 List 生成式
145	第 6 章 函数
146	6.1 认识函数
146	6.1.1 函数如何运行?
147	6.1.2 定义函数
149	6.1.3 调用函数
149	6.1.4 返回值

154	6.2 参数基本机制
154	6.2.1 传递自变量
156	6.2.2 默认参数值
159	6.2.3 关键词参数
161	6.3 巧妙设定参数
161	6.3.1 形参的 * 星号表达式
165	6.3.2 * 运算符拆解可迭代对象
167	6.4 Lambda 函数
169	6.5 变量的适用范围
170	6.5.1 局部变量
171	6.5.2 认识 global 语句
175	第 7 章 模块与函数库
176	7.1 导入模块
176	7.1.1 import/as 语句
176	7.1.2 from/import 语句
177	7.1.3 内置函数 dir() 查看命名空间
178	7.2 自定义模块
178	7.2.1 什么是命令行参数?
181	7.2.2 查看模块路径用 sys.path
181	7.2.3 自行定义模块
182	7.2.4 属性 __name__
184	7.3 随机数值 random 模块
187	7.4 取得时间戳 time 模块
187	7.4.1 取得目前时间
189	7.4.2 时间结构的格式转换
192	7.5 datetime 模块
192	7.5.1 date 类处理日期
196	7.5.2 日期运算有 timedelta 类
198	7.6 显示日历 calendar 模块

203	第 8 章 GUI 界面
204	8.1 浅谈面向对象机制
204	8.1.1 建立类
205	8.1.2 定义方法
206	8.1.3 类实例化
208	8.1.4 先创建再初始化对象
210	8.1.5 有关于继承
211	8.2 使用 tkinter 控件
211	8.2.1 踏出 GUI 第一步
213	8.2.2 建立主窗口
215	8.2.3 tkinter 控件
215	8.2.4 撰写一个简单的窗口程序
217	8.3 控件与版面管理
217	8.3.1 Frame 为容器
218	8.3.2 Button 控件
221	8.3.3 显示文字的标签
223	8.3.4 版面配置 - pack() 方法
229	第 9 章 GUI 其他控件
230	9.1 接收文字的控件
230	9.1.1 接收单行文字的 Entry 控件
231	9.1.2 接收多行文字的 Text 控件
233	9.2 选项控件
233	9.2.1 Checkbutton 控件
236	9.2.2 Radiobutton 控件
237	9.3 绘制图形的 Canvas 控件
237	9.3.1 认识 Canvas 控件
239	9.3.2 绘制几何图形

打造 Python 世界

章节导引	学习目标
1.1 认识 Python	了解 Python 的源起以及它的版本
1.2 青春行——构建 Python	熟悉 Python 的安装和测试
1.3 逛一逛 Python 大观园——IDLE	掌握使用 IDLE 编写程序的方法
1.4 第一个 Python 程序	了解 Python 的编码风格
1.5 新手上路	熟悉新手上路要注意的问题

1.1 认识 Python

Python 程序语言究竟是如何诞生的？对于这个问题，有很多种说法，其中一种说法是，在 1989 年，Python 程序语言的创始人 Guido van Rossum（吉多·范罗苏姆）为了打发圣诞假期，决心为非专业的程序设计人员开发一款新的脚本语言（Script Language），由于他是“蒙提·派森飞行马戏团（Monty Python's Flying Circus）”的爱好者，所以当这款新的脚本语言设计好后，他就以 Python 来命名这款新开发的语言。Python 程序语言自 1989 年推出，至今已有近三十年的历史，它是一款功能强大、成熟且稳定的高级语言，支持命令行编程、函数式编程、面向对象程序设计。同时 Python 程序语言可以跨平台运行，无论是在 Linux、Mac 还是在 Windows 系统上，都可以畅通无阻地使用。

1.1.1 Python 的版本

Python 目前的版本主要包括 2.x 系列和 3.x 系列，下表列举了其发展过程中较为重要的版本。

版本	简介
2.0	2000 年 10 月 16 日发布，支持 Unicode 和垃圾回收机制
2.7.13	2006 年 12 月 17 日发布
3.0	2008 年 12 月 3 日发布，该版本与之前的 Python 2.X 程序不完全兼容
3.5	2015 年 9 月 13 日发布
3.6	2016 年 12 月 23 日发布，也是本书介绍知识点时所采用的版本

一般来说，程序语言会不断以新版本来取代旧版本。而 Python 程序语言的特别之处在于 Python 2.x 和 Python 3.x 同时存在，但彼此之间并不完全兼容。Python 官方声称 Python 2.7 是 Python 2.x 系列的最后版本，该版本可用的资源丰富，有许多第三方函数库都以它为基础。Python 3.0（也称 Python 3000，或简称 Py3k）为了不显得累赘，在设计时没有考虑向下兼容，许多针对早期 Python 版本设计的程序都无法在 Python 3.0 上正常运行。但无论怎样，它们都属于 Python 程序语言。本书将在 Windows 操作系统的的环境下，使用 Python 3.6 来介绍 Python 程序语言的相关语法和结构。

提示 什么是第三方 (Thrid-party) 函数库？

• 为了方便学习者使用，程序语言官方往往会把编写好的程序打包，以“标准函数库” [Standard Library, 也称类别库或模块 (Module)] 的形式供我们使用。Python 必须通过“import”语句导入这些模块才能使用这些程序。

• 第三方函数库 (或称第三方套件) 和第三方模块则是相关的程序开发者编写好的应用程序，它们同样要在 Python 环境下运行，其中包含多种不同的函数，功能强大，应用广泛。

1.1.2 安装 Python 软件

Python 程序语言编写的程序代码必须在 Python 运行环境中进行解释，只有这样系统才能识别这些程序，然后执行程序，输出运行结果。下表介绍了常用的 Python 解释器 (Interpreter)。

解释器	简介
Python	官方提供的免费软件，目前由 Python 软件基金会管理，是官方用 C 语言编写的解释器，本书将使用该软件进行介绍
ZhPy	中文称为“周蟒”，可使用繁/简中文语句编写程序
PyPy	使用 Python 程序语言编写，运行速度比 Python 快
IronPython	可调用 .NET 平台的函数库，将 Python 程序编译成 .NET 程序
Jython	使用 Java 语言编写，可以直接调用 Java 函数库

1.2 青春行——构建 Python 环境

在正式学习 Python 之前，必须先电脑上构建 Python 环境，这样才能在其中编写程序并运行，因此本节将介绍构建 Python 环境的方法。Python 环境的构建主要包括 Python 的下载、安装和测试工作。

1.2.1 下载 Python 软件

首先到 Python 官方网站下载软件。

操作下载 Python 软件

Step 01 进入 Python 官方网站。① 选择“Downloads”选项；② 在下拉菜单中根

据用户使用的操作系统进行选择，此处以“Windows”系统为例，单击“Windows”选项。



Step 02 选择版本“Python 3.6.1”。



Step 03 进入下载页面，滚动页面，找到适用的 Windows 版本（此处下载的版本是 Windows x86 executable installer）。



提示 要安装 Python 的哪一个版本？

Python 提供了多种版本，用户可根据以下分类进行选择。

- x86: 适用于 32 位 Windows 操作系统，包括 Windows XP、Windows 7 和 Windows 8 等。
- x86-64: 适用于 64 位 Windows 操作系统，包括 Windows 10 等。

1.2.2 安装 Python 软件

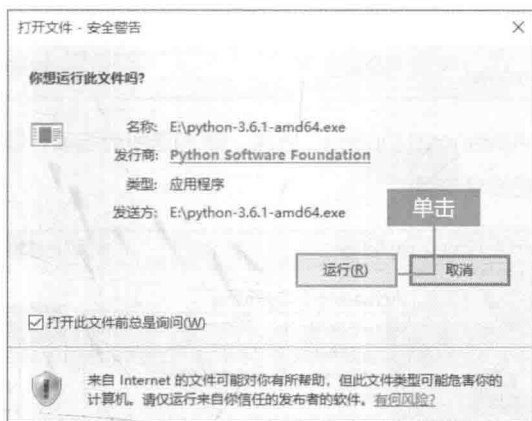
本书使用官方的 Python 软件来介绍相关内容，它包括 Python 3.6 和 pip 两部分。

- Python 3.6: Python 提供的解释器，由 Python 官方团队制作。其源代码完全开放，具有标准架构，任何人都能够根据标准制定 Python 的运行环境，后文中在介绍时会直接用“Python”来指代，不再使用烦琐的版本号。

- pip: 管理 Python 第三方函数库的工具，是 Python 自带的管理工具，可参考下文 Python 软件安装的步骤 4 进行安装。

操作安装 Python 软件

Step 01 双击已经下载好的文件，弹出信息提示框，单击“运行”按钮开始安装。



Step 02 进入软件安装界面。①勾选“Add Python 3.6 to PATH”复选框；②勾选“Install launcher for all users(recommended)”复选框；③选择“Customize installation”选项。

