

# Phthon学习手册：第4版

作者：(美)MarkLutz

O'Reilly精品图书系列

Phthon学习手册

——第4版

Learning Python,Fourth Edition

[美]鲁特兹 (Lutz, M.) 著

李军 刘红伟 等译

ISBN: 978-7-111-32653-3

本书纸版由机械工业出版社于2011年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

## 目 录

[O'Reilly Media, Inc. 介绍](#)  
[译者序](#)  
[前言](#)  
[关于第4版](#)  
[覆盖Python 3.0和Python 2.6](#)  
[新增章](#)  
[已有内容的修改](#)  
[Python 2.6和Python 3.0中的特定语言扩展](#)  
[Python 3.0中特定的语言删除](#)  
[关于本书](#)  
[事前准备](#)  
[本书的范围和其他书籍](#)  
[本书的风格和结构](#)  
[书籍更新](#)  
[关于本书的程序](#)  
[使用代码示例](#)  
[体例](#)  
[联系我们](#)  
[致谢](#)  
[第一部分 使用入门](#)  
[第1章 问答环节](#)  
[人们为何使用Python](#)  
[软件质量](#)  
[开发效率](#)  
[Python是“脚本语言”吗](#)  
[好吧，Python的缺点是什么呢](#)  
[如今谁在使用Python](#)  
[使用Python可以做些什么](#)  
[系统编程](#)  
[用户图形接口](#)  
[Internet脚本](#)  
[组件集成](#)  
[数据库编程](#)  
[快速原型](#)  
[数值计算和科学计算编程](#)  
[游戏、图像、人工智能、XML、机器人等](#)  
[Python如何获得支持](#)  
[Python有哪些技术上的优点](#)  
[面向对象](#)  
[免费](#)  
[可移植](#)  
[功能强大](#)  
[可混合](#)  
[简单易用](#)  
[简单易学](#)  
[Python和其他语言比较起来怎么样](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第2章 Python如何运行程序](#)  
[Python解释器简介](#)  
[程序执行](#)  
[程序员的视角](#)  
[Python的视角](#)  
[执行模块的变体](#)  
[Python实现的替代者](#)  
[执行优化工具](#)  
[冻结二进制文件](#)  
[其他执行选项](#)

[未来的可能性](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第3章 如何运行程序](#)  
[交互提示模式下编写代码](#)  
[交互地运行代码](#)  
[为什么使用交互提示模式](#)  
[使用交互提示模式](#)  
[系统命令行和文件](#)  
[第一段脚本](#)  
[使用命令行运行文件](#)  
[使用命令行和文件](#)  
[UNIX可执行脚本\(!\)](#)  
[点击文件图标](#)  
[在Windows中点击图标](#)  
[input的技巧](#)  
[图标点击的其他限制](#)  
[模块导入和重载](#)  
[模块的显著特性：属性](#)  
[import和reload的使用注意事项](#)  
[使用exec运行模块文件](#)  
[IDLE用户界面](#)  
[IDLE基础](#)  
[使用IDLE](#)  
[高级IDLE工具](#)  
[其他的IDE](#)  
[其他启动选项](#)  
[嵌入式调用](#)  
[冻结二进制的可执行性](#)  
[文本编辑器启动的选择](#)  
[其他的启动选择](#)  
[未来的可能](#)  
[我应该选用哪种](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第一部分 练习题](#)  
[第二部分 类型和运算](#)  
[第4章 介绍Python对象类型](#)  
[为什么使用内置类型](#)  
[Python的核心数据类型](#)  
[数字](#)  
[字符串](#)  
[序列的操作](#)  
[不可变性](#)  
[类型特定的方法](#)  
[寻求帮助](#)  
[编写字符串的其他方法](#)  
[模式匹配](#)  
[列表](#)  
[序列操作](#)  
[类型特定的操作](#)  
[边界检查](#)  
[嵌套](#)  
[列表解析](#)  
[字典](#)  
[映射操作](#)  
[重访嵌套](#)  
[键的排序：for循环](#)

[迭代和优化](#)  
[不存在的键：i测试](#)  
[元组](#)  
[为什么要用元组](#)  
[文件](#)  
[其他文件类工具](#)  
[其他核心类型](#)  
[如何破坏代码的灵活性](#)  
[用户定义的类](#)  
[剩余的内容](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第5章 数字](#)  
[Python的数字类型](#)  
[数字常量](#)  
[内置数学工具和扩展](#)  
[Python表达式操作符](#)  
[在实际应用中的数字](#)  
[变量和基本的表达式](#)  
[数字显示的格式](#)  
[比较：一般的和连续的](#)  
[除法：传统除法、Floor除法和真除法](#)  
[整数精度](#)  
[复数](#)  
[十六进制、八进制和二进制记数](#)  
[位操作](#)  
[其他的内置数学工具](#)  
[其他数字类型](#)  
[小数数字](#)  
[分数类型](#)  
[集合](#)  
[布尔型](#)  
[数字扩展](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第6章 动态类型简介](#)  
[缺少类型声明语句的情况](#)  
[变量、对象和引用](#)  
[类型属于对象，而不是变量](#)  
[对象的垃圾收集](#)  
[共享引用](#)  
[共享引用和在原处修改](#)  
[共享引用和相等](#)  
[动态类型随处可见](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第7章 字符串](#)  
[字符串常量](#)  
[单双引号字符串是一样的](#)  
[用转义序列代表特殊字节](#)  
[raw字符串抑制转义](#)  
[三重引号编写多行字符串块](#)  
[实际应用中的字符串](#)  
[基本操作](#)  
[索引和分片](#)  
[字符串转换工具](#)  
[修改字符串](#)

[字符串方法](#)  
[字符串方法实例：修改字符串](#)  
[字符串方法实例：文本解析](#)  
[实际应用中的其他常见字符串方法](#)  
[最初的字符串模块（在Python 3.0中删除）](#)  
[字符串格式化表达式](#)  
[更高级的字符串格式化表达式](#)  
[基于字典的字符串格式化](#)  
[字符串格式化调用方法](#)  
[基础知识](#)  
[添加键、属性和偏移量](#)  
[添加具体格式化](#)  
[与%格式化表达式比较](#)  
[为什么用新的格式化方法](#)  
[通常意义下的类型分类](#)  
[同样分类的类型共享其操作集合](#)  
[可变类型能够在原处修改](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第8章 列表与字典](#)  
[列表](#)  
[实际应用中的列表](#)  
[基本列表操作](#)  
[列表迭代和解析](#)  
[索引、分片和矩阵](#)  
[原处修改列表](#)  
[字典](#)  
[实际应用中的字典](#)  
[字典的基本操作](#)  
[原处修改字典](#)  
[其他字典方法](#)  
[语言表](#)  
[字典用法注意事项](#)  
[创建字典的其他方法](#)  
[Python 3.0中的字典变化](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第9章 元组、文件及其他](#)  
[元组](#)  
[实际应用中的元组](#)  
[为什么有了列表还要元组](#)  
[文件](#)  
[打开文件](#)  
[使用文件](#)  
[实际应用中的文件](#)  
[其他文件工具](#)  
[重访类型分类](#)  
[对象灵活性](#)  
[引用VS拷贝](#)  
[比较、相等性和真值](#)  
[Python 3.0的字典比较](#)  
[Python中真和假的含义](#)  
[Python的类型层次](#)  
[Type对象](#)  
[Python中的其他类型](#)  
[内置类型陷阱](#)  
[赋值生成引用，而不是拷贝](#)  
[重复能够增加层次深度](#)

[留意循环数据结构](#)  
[不可变类型不可以在原处改变](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第二部分 练习题](#)  
[第三部分 语句和语法](#)  
[第10章 Python语句简介](#)  
[重访Python程序结构](#)  
[Python的语句](#)  
[两个if的故事](#)  
[Python增加了什么](#)  
[Python删除了什么](#)  
[为什么使用缩进语法](#)  
[几个特殊实例](#)  
[简短实例：交互循环](#)  
[一个简单的交互式循环](#)  
[对用户输入数据做数学运算](#)  
[用测试输入数据来处理错误](#)  
[用try语句处理错误](#)  
[嵌套代码三层](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第11章 赋值、表达式和打印](#)  
[赋值语句](#)  
[赋值语句的形式](#)  
[序列赋值](#)  
[Python 3.0中的扩展序列解包](#)  
[多目标赋值语句](#)  
[增强赋值语句](#)  
[变量命名规则](#)  
[表达式语句](#)  
[表达式语句和在原处的修改](#)  
[打印操作](#)  
[Python 3.0的print函数](#)  
[Python 2.6 print语句](#)  
[打印流重定向](#)  
[版本独立的打印](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第12章 if测试和语法规则](#)  
[if语句](#)  
[通用格式](#)  
[基本例子](#)  
[多路分支](#)  
[Python语法规则](#)  
[代码块分隔符](#)  
[语句的分隔符](#)  
[一些特殊情况](#)  
[真值测试](#)  
[if/else三元表达式](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第13章 while和for循环](#)  
[while循环](#)  
[一般格式](#)  
[例子](#)

[break、continue、pass和循环else](#)

[一般循环格式](#)

[pass](#)

[continue](#)

[break](#)

[循环else](#)

[for循环](#)

[一般格式](#)

[例子](#)

[编写循环的技巧](#)

[循环计数器：while和range](#)

[非完备遍历：range和分片](#)

[修改列表：range](#)

[并行遍历：zip和map](#)

[产生偏移和元素：enumerate](#)

[本章小结](#)

[本章习题](#)

[习题解答](#)

[第14章 迭代器和解析，第一部分](#)

[迭代器：初探](#)

[文件迭代器](#)

[手动迭代：iter和next](#)

[其他内置类型迭代器](#)

[列表解析：初探](#)

[列表解析基础知识](#)

[在文件上使用列表解析](#)

[扩展的列表解析语法](#)

[其他迭代环境](#)

[Python 3.0中的新的可迭代对象](#)

[range迭代器](#)

[map、zip和filter迭代器](#)

[多个迭代器VS单个迭代器](#)

[字典视图迭代器](#)

[其他迭代器主题](#)

[本章小结](#)

[本章习题](#)

[习题解答](#)

[第15章 文档](#)

[Python文档资源](#)

[#注释](#)

[dir函数](#)

[文档字符串：\\_\\_doc\\_\\_](#)

[PyDoc：help函数](#)

[PyDoc：HTML报表](#)

[标准手册集](#)

[网络资源](#)

[已出版的书籍](#)

[常见编写代码的陷阱](#)

[本章小结](#)

[本章习题](#)

[习题解答](#)

[第三部分 练习题](#)

[第四部分 函数](#)

[第16章 函数基础](#)

[为何使用函数](#)

[编写函数](#)

[def语句](#)

[def语句是实时执行的](#)

[第一个例子：定义和调用](#)

[定义](#)

[调用](#)  
[Python中的多态](#)  
[第二个例子：寻找序列的交集](#)  
[定义](#)  
[调用](#)  
[重访多态](#)  
[本地变量](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第17章 作用域](#)  
[Python作用域基础](#)  
[作用域法则](#)  
[变量名解析：LEGB原则](#)  
[作用域实例](#)  
[内置作用域](#)  
[global语句](#)  
[最小化全局变量](#)  
[最小化文件间的修改](#)  
[其他访问全局变量的方法](#)  
[作用域和嵌套函数](#)  
[嵌套作用域的细节](#)  
[嵌套作用域举例](#)  
[nonlocal语句](#)  
[nonlocal基础](#)  
[nonlocal应用](#)  
[为什么使用nonlocal](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第18章 参数](#)  
[传递参数](#)  
[参数和共享引用](#)  
[避免可变参数的修改](#)  
[对参数输出进行模拟](#)  
[特定的参数匹配模型](#)  
[基础知识](#)  
[匹配语法](#)  
[细节](#)  
[关键字参数和默认参数的实例](#)  
[任意参数的实例](#)  
[Python 3.0 Keyword-Only参数](#)  
[min调用](#)  
[满分](#)  
[加分点](#)  
[结论](#)  
[一个更有用的例子：通用set函数](#)  
[模拟Python 3.0 print函数](#)  
[使用Keyword-Only参数](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第19章 函数的高级话题](#)  
[函数设计概念](#)  
[递归函数](#)  
[用递归求和](#)  
[编码替代方案](#)  
[循环语句VS递归](#)  
[处理任意结构](#)  
[函数对象：属性和注解](#)



[间接函数调用](#)  
[函数内省](#)  
[函数属性](#)  
[Python 3.0中的函数注解](#)  
[匿名函数: lambda](#)  
[lambda表达式](#)  
[为什么使用lambda](#)  
[如何\(不要\)让Python代码变得晦涩难懂](#)  
[嵌套lambda和作用域](#)  
[在序列中映射函数: map](#)  
[函数式编程工具: filter和reduce](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第20章 迭代和解析, 第二部分](#)  
[回顾列表解析: 函数式编程工具](#)  
[列表解析与map](#)  
[增加测试和嵌套循环](#)  
[列表解析和矩阵](#)  
[理解列表解析](#)  
[重访迭代器: 生成器](#)  
[生成器函数: yield VS return](#)  
[生成器表达式: 迭代器遇到列表解析](#)  
[生成器函数VS生成器表达式](#)  
[生成器是单迭代器对象](#)  
[用迭代工具模拟zip和map](#)  
[内置类型和类中的值生成](#)  
[Python 3.0解析语法概括](#)  
[解析集合和字典解析](#)  
[针对集合和字典的扩展的解析语法](#)  
[对迭代的各种方法进行计时](#)  
[对模块计时](#)  
[计时脚本](#)  
[计时结果](#)  
[计时模块替代方案](#)  
[其他建议](#)  
[函数陷阱](#)  
[本地变量是静态检测的](#)  
[默认和可变对象](#)  
[没有return语句的函数](#)  
[嵌套作用域的循环变量](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第四部分 练习题](#)  
[第五部分 模块](#)  
[第21章 模块: 宏伟蓝图](#)  
[为什么使用模块](#)  
[Python程序架构](#)  
[如何组织一个程序](#)  
[导入和属性](#)  
[标准库模块](#)  
[import如何工作](#)  
[1.搜索](#)  
[2.编译\(可选\)](#)  
[3.运行](#)  
[模块搜索路径](#)  
[配置搜索路径](#)  
[搜索路径的变动](#)  
[sys.path列表](#)

[模块文件选择](#)  
[高级的模块选择概念](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第22章 模块代码编写基础](#)  
[模块的创建](#)  
[模块的使用](#)  
[import语句](#)  
[from语句](#)  
[from\\*语句](#)  
[导入只发生一次](#)  
[import和from是赋值语句](#)  
[文件间变量名的改变](#)  
[import和from的对等性](#)  
[from语句潜在的陷阱](#)  
[模块命名空间](#)  
[文件生成命名空间](#)  
[属性名的点号运算](#)  
[导入和作用域](#)  
[命名空间的嵌套](#)  
[重载模块](#)  
[reload基础](#)  
[reload实例](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第23章 模块包](#)  
[包导入基础](#)  
[包和搜索路径设置](#)  
[\\_\\_init\\_\\_.py包文件](#)  
[包导入实例](#)  
[包对应的from语句和import语句](#)  
[为什么要使用包导入](#)  
[三个系统的传说](#)  
[包相对导入](#)  
[Python 3.0中的变化](#)  
[相对导入基础知识](#)  
[为什么使用相对导入](#)  
[相对导入的作用域](#)  
[模块查找规则总结](#)  
[相对导入的应用](#)  
[本章小结](#)  
[本章习题](#)  
[习题解答](#)  
[第24章 高级模块话题](#)  
[在模块中隐藏数据](#)  
[最小化from\\*的破坏：\\_\\_X\\_\\_和\\_\\_all\\_\\_](#)  
[启用以后的语言特性](#)  
[混合用法模式：\\_\\_name\\_\\_ 和 \\_\\_main\\_\\_](#)  
[以\\_\\_name\\_\\_进行单元测试](#)  
[使用带有\\_\\_name\\_\\_的命令行参数](#)  
[修改模块搜索路径](#)  
[Import语句和from语句的as扩展](#)  
[模块是对象：元程序](#)  
[用名称字符串导入模块](#)  
[过渡性模块重载](#)  
[模块设计理念](#)  
[模块陷阱](#)  
[顶层代码的语句次序的重要性](#)