

Learning Python

原书第5版(下册)
更新Python 2.7和3.3

Python

学习手册



O'REILLY®

 机械工业出版社
China Machine Press


HZ BOOKS
华章IT

Mark Lutz 著
秦鹤 林明 译
林涵菲 审校

原书第5版

Python学习手册

(下册)



Mark Lutz 著

秦鹤 林明 译

林涵菲 审校

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社

图书在版编目 (CIP) 数据

Python 学习手册：原书第 5 版 / (美) 马克·卢茨 (Mark Lutz) 著；秦鹤，林明译。
—北京：机械工业出版社，2018.8

(O'Reilly 精品图书系列)

书名原文：Learning Python, Fifth Edition

ISBN 978-7-111-60366-5

I. P… II. ①马… ②秦… ③林… III. 软件工具—程序设计—手册 IV.TP311.561-62

中国版本图书馆 CIP 数据核字 (2018) 第 146242 号

北京市版权局著作权合同登记

图字：01-2013-5994 号

©2013 Mark Lutz. All rights reserved.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2018. Authorized translation of the English edition, 2018 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2013。

简体中文版由机械工业出版社出版 2018。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

封底无防伪标均为盗版

本书法律顾问

北京大成律师事务所 韩光 / 邹晓东

书 名 / Python 学习手册 (原书第 5 版)

书 号 / ISBN 978-7-111-60366-5

责任编辑 / 陈佳媛

封面设计 / Randy Comer, 张健

出版发行 / 机械工业出版社

地 址 / 北京市西城区百万庄大街 22 号 (邮政编码 100037)

印 刷 / 北京市兆成印刷有限责任公司

开 本 / 178 毫米 × 233 毫米 16 开本 45 印张

版 次 / 2018 年 11 月第 1 版 2018 年 11 月第 1 次印刷

定 价 / 219.00 元 (共 2 册)

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010)88379426；88361066

购书热线：(010)68326294；88379649；68995259

投稿热线：(010)88379604

读者信箱：hzit@hzbook.com

目录

下册

第六部分 类和面向对象编程 (OOP)

第26章 OOP：宏伟蓝图	771
为何使用类.....	772
概览OOP.....	773
属性继承搜索.....	773
类和实例.....	775
方法调用.....	776
编写类树.....	776
运算符重载.....	778
OOP是关于代码重用.....	779
本章小结.....	782
本章习题.....	782
习题解答.....	783
第27章 类代码编写基础	784
类生成多个实例对象.....	784
类对象提供默认行为.....	785
实例对象是具体的元素.....	785
第一个示例.....	786
类通过继承进行定制.....	788
第二个示例.....	789

类是模块内的属性	790
类可以截获Python运算符	791
第三个示例	793
为什么要使用运算符重载	795
世界上最简单的Python类	795
重访记录：类vs字典	798
本章小结	800
本章习题	801
习题解答	801
第28章 一个更加实际的示例	803
步骤1：创建实例	804
编写构造函数	804
在行进中测试	805
以两种方式使用代码	806
版本兼容性提示：Prints	807
步骤2：添加行为方法	808
编写方法	810
步骤3：运算符重载	812
提供打印显示	812
步骤4：通过编写子类定制行为	814
编写子类	814
扩展方法：不好的方式	815
扩展方法：好的方式	815
如果使用super会怎么样呢	817
多态的应用	818
继承、定制和扩展	819
OOP：宏观理念	819
步骤5：定制构造函数	820
OOP比你认为的要简单	821
组合类的其他方式	822
在Python 3.X中捕获内置属性	825

步骤6：使用内省工具	826
特殊的类属性	826
一种通用显示工具	828
实例vs类属性	829
工具类的命名要求	830
类的最终形式	831
步骤7（最后一步）：把对象存储到数据库中	832
pickle和shelve	833
在shelve数据库中存储对象	834
交互式地探索shelve	835
更新shelve中的对象	837
未来方向	838
本章小结	840
本章习题	841
习题解答	841

第29章 类代码编写细节 843

class语句	843
一般形式	844
示例	844
方法	846
示例	847
调用父类构造函数	848
其他方法调用的可能性	848
继承	849
属性树的构造	849
定制被继承的方法	850
类接口技术	851
抽象父类	852
命名空间：结论	855
简单名称：如果被赋值就不是全局的	855
属性名称：对象命名空间	856

Python命名空间之“禅”：赋值决定名称	856
嵌套的类：重温LEGB作用域规则	859
命名空间字典：复习	861
命名空间链接：一个类树爬升器	863
重访文档字符串	865
类vs模块	866
本章小结	867
本章习题	867
习题解答	868
第30章 运算符重载	869
基础知识	869
构造函数和表达式：__init__和__sub__	870
常见的运算符重载方法	870
索引和分片：__getitem__和__setitem__	873
拦截分片	873
Python 2.X中的分片和索引	875
Python 3.X中的__index__不是索引	876
索引迭代：__getitem__	876
可迭代对象：__iter__和__next__	877
用户定义的可迭代对象	878
单个对象上的多个迭代器	881
编程备选方案：__iter__加yield	884
成员关系：__contains__、__iter__和__getitem__	888
属性访问：__getattr__和__setattr__	891
属性引用	891
属性赋值和删除	892
其他属性管理工具	894
模拟实例属性的私有化：第一部分	894
字符串显示：__repr__和__str__	895
为什么需要两种显示方法	896
显示用例提要	898

右侧加法和原位置加法：__radd__和__iadd__	899
右侧加法	899
原位置加法	903
调用表达式：__call__	904
函数接口和基于回调的代码	905
比较运算：__lt__、__gt__和其他方法	907
Python 2.X中的__cmp__方法	908
布尔测试：__bool__和__len__	909
Python 2.X中的布尔方法	910
对象析构函数：__del__	911
析构函数使用提要	912
本章小结	913
本章习题	913
习题解答	913

第31章 类的设计 915

Python和OOP	915
多态意味着接口，不是函数调用签名	916
OOP和继承：“is-a”关系	917
OOP和组合：“has-a”关系	918
重访流处理器	920
请留意：类和持久化	922
OOP和委托：“包装器”代理对象	924
类的伪私有属性	925
名称重整概览	926
为什么使用伪私有属性	927
方法是对象：绑定或未绑定	929
在Python 3.X中，未绑定方法是函数	931
绑定方法和其他可调用对象	932
请留意：绑定方法的回调函数	934
类是对象：一般对象的工厂	935
为什么使用工厂	936

多继承：“mix-in”类	937
编写mix-in显示类	938
与设计相关的其他话题	957
本章小结	957
本章习题	958
习题解答	958
第32章 类的高级主题.....	959
扩展内置类型	960
通过内嵌方式扩展类型	960
通过子类扩展类型	961
“新式”类模型	963
新式类“新”在哪里	964
新式类变化	965
内置属性的获取将跳过实例	966
类型模型改变	971
所有对象派生自“object”	974
钻石继承改变	976
深入讨论MRO：方法解析顺序	980
示例：将属性映射到对应的继承源	983
新式类扩展	988
slot：属性声明	989
property：属性访问器	998
__getattribute__和描述符：属性工具	1001
其他的类改变和扩展	1002
静态方法和类方法	1002
为什么使用特殊方法	1003
Python 2.X和Python 3.X中的静态方法	1003
静态方法备选方案	1005
使用静态方法和类方法	1007
用静态方法计数实例	1008
用类方法计数实例	1009

装饰器和元类：第一部分	1012
函数装饰器基础	1013
初探用户定义函数装饰器	1015
初探类装饰器和元类	1016
更多的细节	1018
super内置函数：更好还是更糟	1018
super引发的大论战	1019
传统的父类调用形式：可移植、通用	1020
super的基础用法与折中	1021
super的优势：继承树的修改与分发	1026
运行时类修改与super	1027
协同多继承方法分发	1028
super总结	1039
类陷阱	1041
修改类属性可能会造成副作用	1041
修改可变类属性也可能产生副作用	1042
多继承：顺序很重要	1043
方法和类中的作用域	1044
其他类陷阱	1045
重温KISS法则：“过度包装”	1046
本章小结	1047
本章习题	1047
习题解答	1047
第六部分练习题	1049
请留意：大师眼中的OOP	1053

第七部分 异常和工具

第33章 异常基础	1057
为什么使用异常	1058
异常的角色	1058
异常：梗概	1059

默认异常处理器	1059
捕获异常	1060
引发异常	1061
用户定义的异常	1062
终止动作	1063
请注意：错误检查	1064
本章小结	1065
本章习题	1066
习题解答	1066
第34章 异常编写细节	1067
try/except/else语句	1067
try语句的工作方式	1068
try语句分句	1069
try else分句	1072
示例：默认行为	1072
示例：捕获内置异常	1074
try/finally语句	1074
示例：利用try/finally编写终止行为	1075
合并try/except/finally语句	1076
合并try语句的语法	1077
通过嵌套合并finally和except	1078
合并try语句的示例	1079
raise语句	1080
引发异常	1081
作用域和try except变量	1082
利用raise传递异常	1084
Python 3.X异常链：raise from	1084
assert语句	1086
示例：捕获约束条件（但不是错误）	1087
with/as上下文管理器	1088
基本用法	1088

上下文管理协议	1090
Python 3.1、2.7及之后版本的多重上下文管理器	1091
本章小结	1093
本章习题	1094
习题解答	1094
第35章 异常对象	1095
异常：回到未来	1096
字符串异常出局了	1096
基于类的异常	1097
编写异常类	1097
为什么使用异常层次结构	1099
内置异常类	1102
内置异常类别	1103
默认打印和状态	1104
定制的打印显示	1106
定制的数据和行为	1107
提供异常细节	1108
提供异常方法	1108
本章小结	1110
本章习题	1110
习题解答	1110
第36章 异常的设计	1112
嵌套异常处理程序	1112
示例：控制流嵌套	1113
示例：语法嵌套化	1114
异常的习惯用法	1116
跳出多重嵌套循环：“go to”	1116
异常并不总是错误	1117
函数可以用raise语句触发状况	1117
断开文件和服务器连接	1118

使用外层try语句辅助调试.....	1119
运行进程中的测试	1120
关于sys.exc_info的更多内容.....	1121
显示错误和跟踪	1121
异常设计的建议与陷阱	1122
应该包装什么	1123
捕获了太多：避免空的except分句和Exception异常.....	1123
捕获过少：使用基于类的异常分类	1125
核心语言总结	1126
Python工具集	1126
大型项目的开发工具.....	1127
本章小结	1130
本章习题	1131
习题解答	1131
第七部分练习题	1131

第八部分 高级主题

第37章 Unicode和字节串	1135
Python 3.X中的字符串修改.....	1136
字符串基础知识	1137
字符编码方案	1137
Python是如何在内存中存储字符串的.....	1139
Python的字符串类型.....	1141
文本和二进制文件	1143
编写基本的字符串.....	1144
Python 3.X的字符串字面量.....	1144
Python 2.X的字符串字面量.....	1146
字符串类型转换	1147
编写Unicode字符串	1148
编写ASCII文本	1148
编写非ASCII文本.....	1149

编码和解码非ASCII文本	1150
其他编码方案	1151
字节串字面量：编码的文本	1152
转换编码	1154
在Python 2.X 中编写Unicode字符串	1154
源文件字符集编码声明	1157
使用Python 3.X bytes对象	1159
方法调用	1159
序列运算	1160
创建bytes对象的其他方式	1161
混合字符串类型	1161
使用Python 3.X/2.6+的 bytearray对象	1162
实际使用中的bytearray	1163
Python 3.X字符串类型小结	1165
使用文本文件和二进制文件	1165
文本文件基础	1166
Python 2.X和3.X中的文本和二进制模式	1166
Python 3.X中类型和内容不匹配	1168
使用Unicode文件	1169
在Python 3.X中读写Unicode	1169
处理Python 3.X中的BOM	1171
Python 2.X中的Unicode文件	1174
Unicode 文件名和流	1175
Python 3.X中其他字符串工具修改	1176
re模式匹配模块	1176
struct二进制数据模块	1177
pickle对象序列化模块	1179
XML解析工具	1181
请留意：检验文件及更多	1184
本章小结	1185
本章习题	1185
习题解答	1186

第38章 被管理的属性	1188
为什么需要被管理属性	1188
插入在属性被访问时运行的代码	1189
property	1190
基础知识	1190
第一个示例.....	1191
计算出的属性.....	1192
使用装饰器编写property	1193
描述符	1195
基础知识	1195
第一个示例.....	1198
计算出的属性.....	1200
在描述符中使用状态信息	1201
property和描述符之间的关系.....	1204
__getattr__和__getattribute__	1206
基础知识	1207
第一个示例.....	1210
计算出的属性.....	1212
比较__getattr__和__getattribute__	1213
管理技术比较.....	1214
拦截内置运算属性	1217
示例：属性验证	1224
使用property验证	1225
使用描述符验证	1227
使用__getattr__验证	1231
使用__getattribute__验证.....	1233
本章小结	1234
本章习题	1234
习题解答	1235

第39章 装饰器	1237
什么是装饰器.....	1237
管理调用和实例.....	1238
管理函数和类.....	1238
使用和定义装饰器.....	1239
为什么使用装饰器.....	1239
基础知识.....	1240
函数装饰器.....	1240
类装饰器.....	1244
装饰器嵌套.....	1247
装饰器参数.....	1249
装饰器管理函数和类.....	1250
编写函数装饰器.....	1250
跟踪调用.....	1250
装饰器状态保持方案.....	1252
类错误一：对方法进行装饰.....	1256
对调用计时.....	1262
添加装饰器参数.....	1265
编写类装饰器.....	1268
单例类.....	1268
跟踪对象接口.....	1270
类错误二：保持多个实例.....	1274
装饰器vs管理器函数.....	1276
为什么使用装饰器（回顾）.....	1277
直接管理函数和类.....	1278
示例：“私有”和“公有”属性.....	1281
实现私有属性.....	1281
实现细节一.....	1283
公有声明的推广.....	1285
实现细节二.....	1287
开放问题.....	1288

Python不是关于控制	1296
示例：验证函数参数	1296
目标	1296
针对位置参数的一个基本范围测试装饰器	1297
关键字和默认参数的推广	1299
实现细节	1302
开放问题	1304
装饰器参数vs函数注解	1306
其他应用：类型测试（如果你坚持）	1308
本章小结	1310
本章习题	1310
习题解答	1311

第40章 元类 1320

勇敢面对还是知难而退	1321
提高“魔法”等级	1322
一门充满钩子的语言	1323
“帮助”函数的缺点	1324
元类vs类装饰器：第一回合	1326
要有魔法，于是就出现了魔法	1328
元类模型	1328
类是类型的实例	1328
元类是Type的子类	1331
Class语句协议	1332
声明元类	1333
在Python 3.X中声明	1333
在Python 2.X中声明	1333
Python 3.X和2.X中的元类分发	1334
编写元类	1335
一个基础的元类	1335
定制构建和初始化	1336
其他元类编程技巧	1337