

**УДК 004.438Python:004.6**  
**ББК 32.973.22**  
**P21**

P21 Лучано Рамальо

Python. К вершинам мастерства / Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2016. – 768 с.: ил.

**ISBN 978-5-97060-384-0**

Язык Python настолько прост, что научиться продуктивно писать на нем программы можно быстро, но зачастую вы при этом используете не все имеющиеся в нем возможности. Данная книга покажет, как создавать эффективный идиоматичный код на Python, задействуя его лучшие – и иногда несправедливо игнорируемые – черты. Автор, Лучано Рамальо, рассказывает о базовых средствах и библиотеках Python и демонстрирует, как сделать код одновременно короче, быстрее и понятнее. Многие опытные программисты стараются подогнать Python под приемы, знакомые им по работе с другими языками. Эта книга покажет, как достичь истинного профессионализма в программировании на Python 3.

Издание предназначено для программистов, уже работающих на Python, но также может быть полезно и начинающим пользователям языка.

УДК 004.438Python:004.6  
 ББК 32.973.22

Original English language edition published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Copyright © 2015 O'Reilly Media, Inc. Russian-language edition copyright © 2015 by DMK Press. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1-491-94600-8 (англ.)  
 ISBN 978-5-97060-384-0 (рус.)

© Luciano Gama de Sousa Ramalho, 2015.  
 © Оформление, перевод на русский язык,  
 издание, ДМК Пресс, 2016



# ОГЛАВЛЕНИЕ

<b>Предисловие .....</b>	<b>17</b>
На кого рассчитана эта книга .....	18
На кого эта книга не рассчитана .....	18
Как организована эта книга .....	18
Практикум .....	20
Как производился хронометраж .....	21
Поговорим: мое личное мнение .....	21
Терминология Python .....	22
Использованная версия Python .....	22
Графические выделения .....	22
О примерах кода .....	23
Как с нами связаться .....	23
Благодарности .....	24
<b>ЧАСТЬ I. Пролог .....</b>	<b>27</b>
<b>Глава 1. Модель данных в языке Python .....</b>	<b>28</b>
Колода карт на Python .....	29
Как используются специальные методы .....	33
Эмуляция числовых типов .....	33
Строковое представление .....	35
Арифметические операторы .....	36
Булево значение пользовательского типа .....	37
Сводка специальных методов .....	37
Почему len – не метод .....	39
Резюме .....	40
Дополнительная литература .....	40
<b>ЧАСТЬ II. Структуры данных .....</b>	<b>43</b>
<b>Глава 2. Массив последовательностей .....</b>	<b>44</b>
Общие сведения о встроенных последовательностях .....	45
Списковое включение и генераторные выражения .....	46
Списковое включение и удобочитаемость .....	46
Сравнение спискового включения с map и filter .....	48
Декартовы произведения .....	49

Генераторные выражения.....	50
Кортеж – не просто неизменяемый список .....	52
Кортежи как записи .....	52
Распаковка кортежа .....	53
Использование * для выборки лишних элементов .....	54
Распаковка вложенного кортежа .....	55
Именованные кортежи .....	56
Кортежи как неизменяемые списки .....	57
Получение среза.....	59
Почему в срезы и диапазоны не включается последний элемент.....	59
Объекты среза .....	59
Многомерные срезы и многоточие .....	61
Присваивание срезу .....	61
Использование + и * для последовательностей .....	62
Построение списка списков .....	63
Составное присваивание последовательностей .....	64
Головоломка: присваивание A += .....	66
Метод list.sort и встроенная функция sorted .....	68
Средства работы с упорядоченными последовательностями в модуле bisect .....	70
Поиск средствами bisect .....	70
Вставка с помощью функции bisect.insort .....	73
Когда список не подходит .....	74
Массивы .....	74
Представления областей памяти.....	78
Библиотеки NumPy и SciPy .....	79
Двусторонние и другие очереди.....	81
Резюме .....	85
Дополнительная литература .....	86
<b>Глава 3. Словари и множества .....</b>	<b>91</b>
Общие типы отображений .....	91
Словарное включение.....	94
Обзор наиболее употребительных методов отображений.....	94
Обработка отсутствия ключей с помощью.setdefault.....	97
Отображения с гибким поиском по ключу .....	99
defaultdict: еще один подход к обработке отсутствия ключа.....	99
Метод __missing__ .....	101
Вариации на тему dict .....	103
Создание подкласса UserDict .....	105
Неизменяемые отображения .....	106
Теория множеств .....	108
Литеральные множества .....	109
Множественное включение .....	111

Операции над множествами.....	111
Под капотом dict и set .....	114
Экспериментальная демонстрация производительности .....	115
Хэш-таблицы в словарях .....	117
Практические последствия механизма работы dict .....	120
Как работают множества – практические следствия.....	123
Резюме.....	123
Дополнительная литература .....	124
Поговорим.....	124
<b>Глава 4. Текст и байты.....</b>	<b>126</b>
О символах и не только .....	127
Все, что нужно знать о байтах .....	128
Структуры и представления областей памяти .....	131
Базовые кодировщики и декодировщики.....	132
Проблемы кодирования и декодирования.....	134
Обработка UnicodeEncodeError .....	134
Обработка UnicodeDecodeError .....	135
Исключение SyntaxError при загрузке модулей и неожиданной кодировкой .....	136
Как определить кодировку последовательности байтов .....	138
BOM: полезный крокозябр .....	139
Обработка текстовых файлов.....	140
Кодировки по умолчанию: сумасшедший дом .....	143
Нормализация Unicode для правильного сравнения .....	146
Сворачивание регистра .....	149
Служебные функции для сравнения нормализованного текста.....	150
Экстремальная «нормализация»: удаление диакритических знаков .....	151
Сортировка Unicode-текстов .....	154
Сортировка с помощью алгоритма упорядочивания Unicode.....	156
База данных Unicode.....	157
Двухрежимный API.....	159
str и bytes в регулярных выражениях .....	159
str и bytes в функциях из модуля os.....	160
Резюме .....	162
Дополнительная литература .....	164
Поговорим.....	166
<b>ЧАСТЬ III. Функции как объекты.....</b>	<b>169</b>
<b>Глава 5. Полноправные функции .....</b>	<b>170</b>
Обращение с функцией как с объектом.....	171
Функции высшего порядка.....	172
Современные альтернативы функциям map, filter и reduce .....	173

Анонимные функции .....	175
Семь видов вызываемых объектов.....	176
Пользовательские вызываемые типы.....	177
Интроспекция функций.....	178
От позиционных к чисто именованным параметрам.....	180
Получение информации о параметрах.....	182
Аннотации функций .....	186
Пакеты для функционального программирования .....	188
Модуль operator .....	188
Фиксация аргументов с помощью functools.partial.....	191
Резюме.....	193
Дополнительная литература .....	194
Поговорим.....	195

## **Глава 6. Реализация паттернов проектирования с помощью полноправных функций ..... 198**

Практический пример: переработка паттерна Стратегия .....	199
Классическая Стратегия .....	199
Функционально-ориентированная стратегия.....	203
Выбор наилучшей стратегии: простой подход.....	206
Поиск стратегий в модуле .....	207
Паттерн Команда .....	208
Резюме.....	210
Дополнительная литература .....	211
Поговорим.....	212

## **Глава 7. Декораторы функций и замыкания ..... 214**

Краткое введение в декораторы .....	215
Когда Python выполняет декораторы.....	216
Паттерн Стратегия, дополненный декоратором.....	218
Правила видимости переменных .....	219
Замыкания.....	222
Объявление nonlocal.....	225
Реализация простого декоратора .....	227
Как это работает .....	228
Декораторы в стандартной библиотеке .....	230
Кэширование с помощью functools.lru_cache .....	230
Одиночная диспетчеризация и обобщенные функции .....	233
Композиции декораторов .....	236
Параметризованные декораторы.....	236
Параметризованный регистрационный декоратор.....	237
Параметризованный декоратор clock.....	239

Резюме .....	242
Дополнительная литература .....	242
Поговорим .....	243

## **ЧАСТЬ IV. Объектно-ориентированные идиомы..... 247**

### **Глава 8. Ссылки на объекты, изменяемость и повторное использование ..... 248**

Переменные – не ящики .....	249
Тождественность, равенство и синонимы .....	250
Выбор между == и is .....	252
Относительная неизменяемость кортежей.....	253
По умолчанию копирование поверхностное .....	254
Глубокое и поверхностное копирование произвольных объектов .....	256
Параметры функций как ссылки .....	258
Значения по умолчанию изменяемого типа: неудачная мысль.....	259
Защитное программирование при наличии изменяемых параметров .....	261
del и сборка мусора .....	263
Слабые ссылки .....	265
Коллекция WeakValueDictionary .....	266
Ограничения слабых ссылок.....	268
Как Python хитрит с неизменяемыми объектами .....	269
Резюме .....	270
Дополнительная литература .....	271
Поговорим.....	272

### **Глава 9. Объект в духе Python ..... 276**

Представления объекта .....	277
И снова класс вектора .....	277
Альтернативный конструктор .....	280
Декораторы classmethod и staticmethod.....	281
Форматирование при выводе .....	282
Хэшируемый класс Vector2d .....	286
Закрытые и «защищенные» атрибуты в Python .....	291
Экономия памяти с помощью атрибута класса __slots__ .....	293
Проблемы при использовании __slots__ .....	296
Переопределение атрибутов класса .....	296
Резюме.....	299
Дополнительная литература .....	300
Поговорим.....	301

### **Глава 10. Рубим, перемешиваем и нарезаем последовательности ..... 305**

Vector: пользовательский тип последовательности.....	306
Vector, попытка № 1: совместимость с Vector2d .....	306
Протоколы и динамическая типизация.....	309
Vector, попытка № 2: последовательность, допускающая срезку .....	310
Как работает срезка .....	311
Метод __getitem__ с учетом срезов .....	313
Vector, попытка № 3: доступ к динамическим атрибутам .....	315
Vector, попытка № 4: хэширование и ускорение оператора ==.....	319
Vector, попытка № 5:	
форматирование .....	324
Резюме.....	331
Дополнительная литература .....	332
Поговорим.....	333

## **Глава 11. Интерфейсы: от протоколов до абстрактных базовых классов..... 338**

Интерфейсы и протоколы в культуре Python.....	339
Python в поисках следов последовательностей.....	341
Партизанское латание как средство реализации протокола во время выполнения.....	343
Алекс Мартелли о водоплавающих .....	345
Создание подкласса ABC.....	350
ABC в стандартной библиотеке.....	352
ABC в модуле collections.abc .....	352
Числовая башня ABC.....	354
Определение и использование ABC.....	355
Синтаксические детали ABC.....	359
Создание подклассов ABC Tombola.....	360
Виртуальный подкласс Tombola .....	363
Как тестировались подклассы Tombola .....	365
Использование метода register на практике .....	368
Гуси могут вести себя как утки .....	369
Резюме.....	371
Дополнительная литература .....	373
Поговорим.....	374

## **Глава 12. Наследование: хорошо или плохо ..... 380**

Сложности наследования встроенным типам .....	380
Множественное наследование и порядок разрешения методов .....	384
Множественное наследование в реальном мире .....	388
Жизнь с множественным наследованием .....	391
Tkinter: хороший, плохой, злой .....	393

Современный пример: примеси в обобщенных представлениях	
Django .....	395
Резюме .....	398
Дополнительная литература .....	399
Поговорим .....	400

## **Глава 13. Перегрузка операторов: как правильно? ..... 403**

Основы перегрузки операторов .....	404
Унарные операторы .....	404
Перегрузка оператора сложения векторов + .....	407
Перегрузка оператора умножения на скаляр * .....	412
Операторы сравнения .....	416
Операторы составного присваивания .....	420
Резюме .....	425
Дополнительная литература .....	426
Поговорим .....	427

## **ЧАСТЬ V. Поток управления ..... 431**

### **Глава 14. Итерируемые объекты, итераторы и генераторы .. 432**

Класс Sentence, попытка № 1: последовательность слов .....	433
Почему последовательности итерируемы: функция iter .....	435
Итерируемые объекты и итераторы .....	436
Класс Sentence, попытка № 2: классический вариант .....	440
Почему идея сделать Sentence итератором плоха .....	442
Класс Sentence, попытка № 3: генераторная функция .....	443
Как работает генераторная функция .....	444
Класс Sentence, попытка № 4: ленивая реализация .....	447
Класс Sentence, попытка № 5: генераторное выражение .....	448
Генераторные выражения: когда использовать .....	450
Другой пример: генератор арифметической прогрессии .....	451
Построение арифметической прогрессии с помощью itertools .....	453
Генераторные функции в стандартной библиотеке .....	454
yield from – новая конструкция в Python 3.3 .....	465
Функции редуцирования итерируемого объекта .....	466
Более пристальный взгляд на функцию iter .....	468
Пример: генераторы в утилите преобразования базы данных .....	469
Генераторы как сопрограммы .....	471
Резюме .....	472
Дополнительная литература .....	472
Поговорим .....	473



<b>Глава 15. Контекстные менеджеры и блоки else .....</b>	<b>479</b>
Делай то, потом это: блоки else вне if .....	480
Контекстные менеджеры и блоки with .....	482
Утилиты contextlib .....	486
Использование @contextmanager .....	487
Резюме .....	490
Дополнительная литература .....	491
Поговорим .....	492
<b>Глава 16. Сопрограммы .....</b>	<b>494</b>
Эволюция: от генераторов к сопрограммам .....	495
Базовое поведение генератора, используемого в качестве сопрограммы .....	496
Пример: сопрограмма для вычисления накопительного среднего .....	499
Декораторы для инициализации сопрограмм .....	501
Завершение сопрограммы и обработка исключений .....	502
Возврат значения из сопрограммы .....	506
Использование yield from .....	508
Семантика yield from .....	514
Пример: применение сопрограмм для моделирования дискретных событий .....	520
О моделировании дискретных событий .....	521
Моделирование работы таксопарка .....	522
Резюме .....	529
Дополнительная литература .....	531
Поговорим .....	533
<b>Глава 17. Параллелизм и будущие объекты .....</b>	<b>536</b>
Пример: три способа загрузки из веба .....	536
Скрипт последовательной загрузки .....	538
Загрузка с применением библиотеки concurrent.futures .....	540
Где находятся будущие объекты? .....	542
Блокирующий ввод-вывод и GiL .....	545
Запуск процессов с помощью concurrent.futures .....	546
Эксперименты с Executor.map .....	548
Загрузка с индикацией хода выполнения и обработкой ошибок .....	551
Обработка ошибок во flags2-примерах .....	556
Использование futures.as_completed .....	558
Альтернативы: многопоточная и многопроцессная обработка .....	561
Резюме .....	561
Дополнительная литература .....	562
Поговорим .....	564

<b>Глава 18. Применение пакета <code>asyncio</code> для организации конкурентной работы</b>	<b>567</b>
Сравнение потока и сопрограммы	569
<code>asyncio.Future</code> : не блокирует умышленно	575
<code>Yield from</code> из будущих объектов, задач и сопрограмм	576
Загрузка с применением <code>asyncio</code> и <code>aiohttp</code>	578
Объезд блокирующих вызовов	582
Улучшение скрипта загрузки на основе <code>asyncio</code>	585
Использование <code>asyncio.as_completed</code>	585
Использование исполнителя для предотвращения блокировки цикла обработки событий	591
От обратных вызовов к будущим объектам и сопрограммам	592
Выполнение нескольких запросов для каждой операции загрузки	595
Разработка серверов с помощью пакета <code>asyncio</code>	597
TCP-сервер на основе <code>asyncio</code>	598
Веб-сервер на основе библиотеки <code>aiohttp</code>	602
Повышение степени параллелизма за счет более интеллектуальных клиентов	606
Резюме	607
Дополнительная литература	608
Поговорим	610
<b>ЧАСТЬ VI. Метaprogramмирование</b>	<b>613</b>
<b>Глава 19. Динамические атрибуты и свойства</b>	<b>614</b>
Применение динамических атрибутов для обработки данных	615
Исследование JSON-подобных данных с динамическими атрибутами	617
Проблема недопустимого имени атрибута	620
Гибкое создание объектов с помощью метода <code>__new__</code>	622
Изменение структуры набора данных OSCON с помощью модуля <code>shelve</code>	624
Выборка связанных записей с помощью свойств	627
Использование свойств для контроля атрибутов	633
<code>Linelltem</code> , попытка № 1: класс строки заказа	633
<code>Linelltem</code> , попытка № 2: контролирующее свойство	634
Правильный взгляд на свойства	636
Свойства переопределяют атрибуты экземпляра	637
Документирование свойств	639
Программирование фабрики свойств	640
Удаление атрибутов	643
Важные атрибуты и функции для работы с атрибутами	644
Специальные атрибуты, влияющие на обработку атрибутов	645
Встроенные функции для работы с атрибутами	645
Специальные методы для работы с атрибутами	646
Резюме	648



Дополнительная литература .....	648
Поговорим .....	649
<b>Глава 20. Дескрипторы атрибутов .....</b>	<b>653</b>
Пример дескриптора: проверка значений атрибутов .....	653
Lineltcm попытка № 3: простой дескриптор .....	654
Lineltcm попытка № 4: автоматическая генерация имен атрибутов хранения .....	659
Lineltcm попытка № 5: новый тип дескриптора .....	665
Переопределяющие и непереопределяющие дескрипторы .....	668
Переопределяющий дескриптор .....	669
Переопределяющий дескриптор без <code>__get__</code> .....	670
Непереопределяющий дескриптор .....	671
Перезаписывание дескриптора в классе .....	673
Методы являются дескрипторами .....	673
Советы по использованию дескрипторов .....	676
Строка документации дескриптора и перехват удаления .....	677
Резюме .....	678
Дополнительная литература .....	679
Поговорим .....	680
<b>Глава 21. Метапрограммирование классов .....</b>	<b>682</b>
Фабрика классов .....	683
Декоратор класса для настройки дескрипторов .....	686
Что когда происходит: этап импорта и этап выполнения .....	688
Демонстрация работы интерпретатора .....	689
Основы метаклассов .....	693
Демонстрация работы метакласса .....	695
Метакласс для настройки дескрипторов .....	699
Специальный метод метакласса <code>__prepare__</code> .....	701
Классы как объекты .....	703
Резюме .....	704
Дополнительная литература .....	705
Поговорим .....	707
<b>Послесловие .....</b>	<b>709</b>
Дополнительная литература .....	710
<b>Приложение А. Основы языка Python .....</b>	<b>713</b>
Глава 3: тест производительности оператора <code>in</code> .....	713
Глава 3: сравнение битовых представлений хэшей .....	715
Глава 9. Потребление оперативной памяти при наличии и отсутствии <code>__slots__</code> .....	716

Глава 14: скрипт преобразования базы данных isis2json.py .....	717
Глава 16: моделирование дискретных событий таксопарка .....	722
Глава 17: примеры, относящиеся к криптографии .....	726
Глава 17: примеры HTTP-клиентов из серии flags2 .....	729
Глава 19: скрипты и тесты для обработки набора данных OSCON .....	734
<b>Терминология Python .....</b>	<b>739</b>
<b>Предметный указатель .....</b>	<b>754</b>