

Python

Лучшие практики и инструменты

ТРЕТЬЕ ИЗДАНИЕ

Михал Яворски, Тарек Зиаде

Pyt
Packt>

 ПИТЕР®

Python

ЛУЧШИЕ ПРАКТИКИ И ИНСТРУМЕНТЫ

Михал Яворски
Тарек Зиаде

ТРЕТЬЕ ИЗДАНИЕ

 ПИТЕР®

Санкт-Петербург · Москва · Минск

2021

ББК 32.973.2-018.1
УДК 004.43
Я22

Яворски Михал, Зиаде Тарек

Я22 Python. Лучшие практики и инструменты. — СПб.: Питер, 2021. — 560 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-1589-1

Python — это динамический язык программирования, используемый в самых разных предметных областях. Хотя писать код на Python просто, гораздо сложнее сделать этот код удобочитаемым, пригодным для многократного использования и легким в поддержке. Третье издание «Python. Лучшие практики и инструменты» даст вам инструменты для эффективного решения любой задачи разработки и сопровождения софта.

Авторы начинают с рассказа о новых возможностях Python 3.7 и продвинутых аспектах синтаксиса Python. Продолжают советами по реализации популярных парадигм, в том числе объектно-ориентированного, функционального и событийно-ориентированного программирования. Также авторы рассказывают о наилучших практиках именовании, о том, какими способами можно автоматизировать развертывание программ на удаленных серверах. Вы узнаете, как создавать полезные расширения для Python на C, C++, Cython и CFFI.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.973.2-018.1
УДК 004.43

Права на издание получены по соглашению с Packt Publishing. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги. Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1789808896 англ.

© Packt Publishing 2019.
First published in the English language under the title 'Expert Python Programming – Third Edition – (9781789808896)'

ISBN 978-5-4461-1589-1

© Перевод на русский язык ООО Издательство «Питер», 2021
© Издание на русском языке, оформление ООО Издательство «Питер», 2021
© Серия «Библиотека программиста», 2021
© Перевод с англ. А. Павлов

Краткое содержание

О создателях книги	14
Предисловие	15
От издательства	20

Часть I. Перед началом работы

Глава 1. Текущее состояние Python	22
Глава 2. Современные среды разработки на Python	39

Часть II. Ремесло Python

Глава 3. Современные элементы синтаксиса — ниже уровня класса	66
Глава 4. Современные элементы синтаксиса — выше уровня класса	123
Глава 5. Элементы метапрограммирования	152
Глава 6. Как выбирать имена	173
Глава 7. Создаем пакеты	195
Глава 8. Развертывание кода	231
Глава 9. Расширения Python на других языках	268

Часть III. Качество, а не количество

Глава 10. Управление кодом	308
Глава 11. Документирование проекта	339
Глава 12. Разработка на основе тестирования	366

Часть IV. Жажда скорости

Глава 13. Оптимизация — принципы и методы профилирования	404
Глава 14. Эффективные методы оптимизации	434
Глава 15. Многозадачность	461

Часть V. Техническая архитектура

Глава 16. Событийно-ориентированное и сигнальное программирование	504
Глава 17. Полезные паттерны проектирования	523
Приложение. reStructuredText Primer	552

Оглавление

О создателях книги	14
Об авторах.....	14
О научном редакторе.....	14
Предисловие	15
Для кого эта книга.....	15
Что мы рассмотрим	16
Как получить максимум от этой книги.....	17
Скачивание файлов с примерами кода.....	18
Скачивание цветных изображений.....	18
Условные обозначения	18
От издательства	20

Часть I. Перед началом работы

Глава 1. Текущее состояние Python	22
Технические требования.....	23
Где мы находимся и куда движемся	23
Почему и как изменился язык Python	23
Как не отставать от изменений в документации PEP	24
Внедрение Python 3 на момент написания этой книги	25
Основные различия между Python 3 и Python 2.....	26
Почему это должно нас волновать	26
Основные синтаксические различия и распространенные ошибки.....	27
Популярные инструменты и методы поддержания кросс-версионной совместимости	29
Не только CPython	33
Почему это должно нас волновать	33
Stackless Python	33
Jython	34
IronPython.....	35
PyPy	36
MicroPython	36
Полезные ресурсы	37
Резюме	38
Глава 2. Современные среды разработки на Python.....	39
Технические требования.....	40
Установка дополнительных пакетов Python с использованием pip.....	40
Изоляция сред выполнения.....	42
venv — виртуальное окружение Python	43
Изоляция среды на уровне системы.....	46
Виртуальные среды разработки, использующие Vagrant.....	47
Виртуальные среды, использующие Docker	49

Популярные инструменты повышения производительности	59
Пользовательские оболочки Python — ipython, bpython, ptpython и т. д.	60
Включение оболочек в собственные скрипты и программы	62
Интерактивные отладчики	63
Резюме	64

Часть II. Ремесло Python

Глава 3. Современные элементы синтаксиса — ниже уровня класса	66
Технические требования	67
Встроенные типы языка Python	67
Строки и байты	67
Контейнеры	73
Дополнительные типы данных и контейнеры	85
Специализированные контейнеры данных из модуля collections	85
Символическое перечисление с модулем enum	86
Расширенный синтаксис	88
Итераторы	88
Генераторы и операторы yield	91
Декораторы	94
Менеджеры контекста и оператор with	105
Функционально-стилевые особенности Python	109
Что такое функциональное программирование	110
Лямбда-функции	111
map(), filter() и reduce()	112
Частичные объекты и функция partial()	115
Выражения генераторов	116
Аннотации функций и переменных	117
Общий синтаксис	117
Возможные способы применения	118
Статическая проверка типа с помощью typing	118
Иные элементы синтаксиса, о которых вы, возможно, не знаете	119
Оператор for... else	119
Именованные аргументы	120
Резюме	122
Глава 4. Современные элементы синтаксиса — выше уровня класса	123
Технические требования	124
Протоколы в языке Python — методы и атрибуты с двойным подчеркиванием	124
Сокращение шаблонного кода с помощью классов данных	126
Создание подклассов встроенных типов	128
ПРМ и доступ к методам из суперклассов	131
Классы старого стиля и суперклассы в Python 2	133
Понимание ПРМ в Python	134
Ловушки суперкласса	138
Практические рекомендации	141
Паттерны доступа к расширенным атрибутам	141
Дескрипторы	142
Свойства	147
Слоты	150
Резюме	151

Глава 5. Элементы метапрограммирования	152
Технические требования.....	152
Что такое метапрограммирование.....	153
Декораторы как средство метапрограммирования.....	153
Декораторы класса	154
Использование <code>__new__()</code> для переопределения процесса создания экземпляра	156
Метаклассы.....	158
Генерация кода.....	165
Резюме	172
Глава 6. Как выбирать имена	173
Технические требования.....	174
PEP 8 и практические рекомендации по именованию.....	174
Почему и когда надо соблюдать PEP 8	174
За пределами PEP 8 — правила стиля внутри команды	175
Стили именования	175
Переменные.....	176
Руководство по именованию	184
Использование префиксов <code>is/has</code> в булевых элементах	184
Использование множественного числа в именах коллекций	185
Использование явных имен для словарей	185
Избегайте встроенных и избыточных имен.....	185
Избегайте уже существующих имен	186
Практические рекомендации по работе с аргументами	187
Сборка аргументов по итеративному принципу	187
Доверие к аргументам и тестам	188
Осторожность при работе с магическими аргументами <code>*args</code> и <code>**kwargs</code>	188
Имена классов	190
Имена модулей и пакетов	191
Полезные инструменты	191
PyLint.....	192
runcodestyle и flake8	193
Резюме	194
Глава 7. Создаем пакеты	195
Технические требования.....	195
Создание пакета	196
Странности в нынешних инструментах создания пакетов в Python	196
Конфигурация проекта.....	198
Пользовательская команда <code>setup</code>	207
Работа с пакетами в процессе разработки.....	208
Пакеты пространства имен	209
Почему это полезно	210
Загрузка пакета	214
PyPI — каталог пакетов Python	214
Пакеты с исходным кодом и пакеты сборок.....	216
Исполняемые файлы.....	220
Когда бывают полезны исполняемые файлы.....	221
Популярные инструменты	221
Безопасность кода Python в исполняемых пакетах	228
Резюме	230

Глава 8. Развертывание кода	231
Технические требования.....	232
Двенадцатифакторное приложение	232
Различные подходы к автоматизации развертывания	234
Использование Fabric для автоматизации развертывания.....	235
Ваш собственный каталог пакетов или зеркало каталогов	239
Зеркала PyPI	240
Объединение дополнительных ресурсов с пакетом Python	241
Общие соглашения и практики	249
Иерархия файловой системы	249
Изоляция	250
Использование инструментов мониторинга процессов	250
Запуск кода приложения в пространстве пользователя.....	252
Использование обратного HTTP-прокси.....	253
Корректная перезагрузка процессов	254
Контрольно-проверочный код и мониторинг	256
Ошибки журнала — Sentry/Raven	256
Метрики систем мониторинга и приложений	260
Работа с журнальными приложениями	262
Резюме	267
Глава 9. Расширения Python на других языках	268
Технические требования.....	269
Различия между языками C и C++	269
Необходимость в использовании расширений	272
Повышение производительности критических фрагментов кода	272
Интеграция существующего кода, написанного на разных языках.....	273
Интеграция сторонних динамических библиотек	274
Создание пользовательских типов данных	274
Написание расширений.....	275
Расширения на чистом языке C	276
Написание расширений на Cython	291
Проблемы с использованием расширений	295
Дополнительная сложность.....	296
Отладка	297
Взаимодействие с динамическими библиотеками без расширений	297
Модуль ctypes	298
CFFI	304
Резюме	306

Часть III. Качество, а не количество

Глава 10. Управление кодом	308
Технические требования.....	308
Работа с системой управления версиями	308
Централизованные системы.....	309
Распределенные системы	312
Распределенные стратегии	313
Централизованность или распределенность.....	314
По возможности используйте Git.....	315
Рабочий процесс GitFlow и GitHub Flow	316

Настройка процесса непрерывной разработки	320
Непрерывная интеграция	321
Непрерывная доставка	325
Непрерывное развертывание	326
Популярные инструменты для непрерывной интеграции	326
Выбор правильного инструмента и распространенные ошибки	335
Резюме	338
Глава 11. Документирование проекта	339
Технические требования	339
Семь правил технической документации	340
Пишите в два этапа	340
Ориентируйтесь на читателя	341
Упрощайте стиль	342
Ограничивайте объем информации	342
Используйте реалистичные примеры кода	343
Пишите по минимуму, но достаточно	344
Используйте шаблоны	344
Документация как код	345
Использование строк документации в Python	345
Популярные языки разметки и стилей для документации	347
Популярные генераторы документации для библиотек Python	348
Sphinx	349
MkDocs	352
Сборка документации и непрерывная интеграция	352
Документирование веб-API	353
Документация как прототип API с API Blueprint	354
Самодокументирующиеся API со Swagger/OpenAPI	355
Создание хорошо организованной системы документации	356
Создание портфеля документации	356
Ваш собственный портфель документации	362
Создание шаблона документации	363
Шаблон для автора	364
Шаблон для читателя	364
Резюме	365
Глава 12. Разработка на основе тестирования	366
Технические требования	366
Я не тестирую	367
Три простых шага разработки на основе тестирования	367
О каких тестах речь	372
Стандартные инструменты тестирования в Python	375
Я тестирую	380
Ловушки модуля unittest	380
Альтернативы модулю unittest	381
Охват тестирования	388
Подделки и болванки	390
Совместимость среды тестирования и зависимостей	396
Разработка на основе документации	400
Резюме	402

Часть IV. Жажда скорости

Глава 13. Оптимизация — принципы и методы профилирования	404
Технические требования.....	404
Три правила оптимизации.....	405
Сначала — функционал	405
Работа с точки зрения пользователя.....	406
Поддержание читабельности и удобства сопровождения	407
Стратегии оптимизации	408
Пробуем свалить вину на другого	408
Масштабирование оборудования	409
Написание теста скорости.....	410
Поиск узких мест	410
Профилерование использования ЦП	411
Профилерование использования памяти.....	419
Профилерование использования сети.....	430
Резюме	433
Глава 14. Эффективные методы оптимизации.....	434
Технические требования.....	435
Определение сложности	436
Цикломатическая сложность	437
Нотация «O большое».....	438
Уменьшение сложности через выбор подходящей структуры данных	440
Поиск в списке.....	440
Использование модуля collections	442
Тип deque	442
Тип defaultdict.....	444
Тип namedtuple	444
Использование архитектурных компромиссов	446
Использование эвристических алгоритмов или приближенных вычислений	446
Применение очереди задач и отложенная обработка.....	447
Использование вероятностной структуры данных	450
Кэширование	451
Детерминированное кэширование	452
Недетерминированное кэширование.....	455
Сервисы кэширования.....	456
Резюме	460
Глава 15. Многозадачность	461
Технические требования.....	461
Зачем нужна многозадачность	462
Многопоточность	463
Что такое многопоточность.....	464
Как Python работает с потоками.....	465
Когда использовать многопоточность	466
Многопроцессорная обработка	481
Встроенный модуль multiprocessing	483
Асинхронное программирование.....	489
Кооперативная многозадачность и асинхронный ввод/вывод.....	490
Ключевые слова async и await.....	491

Модуль <code>asyncio</code> в старых версиях Python	495
Практический пример асинхронного программирования	495
Интеграция синхронного кода с помощью фьючерсов <code>async</code>	498
Резюме	501

Часть V. Техническая архитектура

Глава 16. Событийно-ориентированное и сигнальное программирование	504
Технические требования.....	505
Что такое событийно-ориентированное программирование	505
Событийно-ориентированный != асинхронный.....	506
Событийно-ориентированное программирование в GUI.....	507
Событийно-ориентированная связь	509
Различные стили событийно-ориентированного программирования.....	511
Стиль на основе обратных вызовов.....	511
Стиль на основе субъекта	513
Тематический стиль	515
Событийно-ориентированные архитектуры	518
Очереди событий и сообщений	519
Резюме	521
Глава 17. Полезные паттерны проектирования	523
Технические требования.....	524
Порождающие паттерны.....	524
Синглтон.....	524
Структурные паттерны.....	527
Адаптер	528
Заместитель.....	542
Фасад.....	543
Поведенческие паттерны	544
Наблюдатель	544
Посетитель	546
Шаблонный метод.....	548
Резюме	550
Приложение. <code>reStructuredText Primer</code>	552
<code>reStructuredText</code>	552
Структура раздела	554
Списки	555
Форматирование внутри строк	556
Блок литералов.....	557
Ссылки	558