

# Оглавление

<b>Об авторе.....</b>	<b>12</b>
<b>О научном редакторе .....</b>	<b>13</b>
<b>Благодарности .....</b>	<b>14</b>
<b>Введение .....</b>	<b>15</b>
Кому и зачем следует читать эту книгу.....	16
О книге .....	16
От издательства .....	18
<b>Глава 1. Начало проекта .....</b>	<b>19</b>
Версии Python .....	19
План нового проекта .....	20
Что делать .....	21
Что не делать.....	22
Нумерация версий .....	23
Стиль кода и автоматические проверки.....	24
Инструменты для выявления несоответствия стиля.....	26
Инструменты для выявления ошибок в коде.....	27
Джошуа Харлоу о Python .....	28
<b>Глава 2. Модули, библиотеки и фреймворки .....</b>	<b>31</b>
Система импорта .....	31
Модуль sys .....	33
Импорт пути.....	33

---

Пользовательские импортеры .....	34
Поисковик метапути .....	35
Полезные стандартные библиотеки .....	37
Внешние библиотеки .....	39
Проверка безопасности использования внешних библиотек .....	40
Защита кода с помощью обертки API.....	41
Установка пакетов: получение большего от pip .....	41
Выбор и использование фреймворков .....	44
Разработчик ядра Python Дуг Хелман о библиотеках .....	45
<b>Глава 3. Документация и практики хорошего API .....</b>	<b>51</b>
Документирование со Sphinx .....	51
Начало работы со Sphinx и reST .....	53
Модули Sphinx.....	54
Написание расширения для Sphinx.....	57
Управление изменениями в API .....	59
Нумерация версий API.....	60
Документирование изменений в API.....	60
Обозначение неактуальных функций модулем warnings .....	62
Итоги.....	64
Кристоф де Вьенн о разработке API .....	65
<b>Глава 4. Работа с временными метками и часовыми поясами .....</b>	<b>68</b>
Проблема отсутствующих часовых поясов .....	68
Создание объекта datetime по умолчанию .....	69
Создание временных меток с учетом часового пояса с помощью dateutil .....	71
Сериализация объектов datetime с учетом часового пояса.....	73
Работа с неоднозначным временем .....	75
Итоги.....	76
<b>Глава 5. Распространение ПО.....</b>	<b>77</b>
История setup.py .....	77
Пакетирование с setup.cfg .....	80
Стандарт распространения Wheel.....	82

Как распространить свой проект .....	84
Точки входа.....	88
Визуализация точки входа.....	89
Использование сценариев командной строки .....	90
Использование плагинов и драйверов .....	93
Итоги.....	96
Ник Коглан о пакетировании .....	96
<b>Глава 6. Модульное тестирование .....</b>	<b>98</b>
Основы тестирования .....	98
Простые тесты.....	98
Пропуск тестов.....	101
Запуск определенных тестов .....	102
Параллельный запуск тестов.....	104
Создание объектов, используемых в тестах, с помощью фикстур .....	105
Запуск тестовых сценариев .....	107
Управляемые тесты с объектами-пустышками .....	108
Выявление непротестированного кода с помощью coverage .....	113
Виртуальное окружение .....	115
Настройка виртуального окружения .....	116
Использование virtualenv с tox.....	118
Повторное создание окружения .....	119
Использование других версий Python .....	121
Интеграция с другими тестами .....	121
Политика тестирования .....	122
Роберт Коллинз о тестировании .....	124
<b>Глава 7. Методы и декораторы.....</b>	<b>126</b>
Декораторы и их применение .....	126
Создание декораторов.....	127
Написание декораторов .....	128
Использование нескольких декораторов.....	129
Написание декораторов класса .....	130
Работа методов в Python .....	135

---

Статические методы .....	137
Классовый метод .....	138
Абстрактные методы .....	139
Смесь статического, классического и абстрактного методов .....	141
Включение реализации в абстрактный метод .....	143
Правда о super .....	144
Итоги.....	147
<b>Глава 8. Функциональное программирование.....</b>	<b>148</b>
Создание чистых функций .....	148
Генераторы .....	149
Создание генератора.....	150
Возвращение и передача значения с помощью yield.....	152
inspect и генераторы .....	153
Списковое включение .....	155
Функции функционального стиля .....	156
Применение функций к элементам с помощью map() .....	156
Фильтрация списка с помощью filter() .....	157
Получение индексов с enumerate().....	157
Сортировка списка с помощью sorted().....	158
Поиск элементов по условию с помощью any() или all() .....	158
Комбинирование списков с помощью zip().....	159
Решение распространенных проблем .....	159
Использование lambda() с functools .....	161
Полезные функции itertools .....	163
Итоги.....	164
<b>Глава 9. Абстрактное синтаксическое дерево, диалект Ну и Lisp-образные атрибуты .....</b>	<b>165</b>
Изучение АСД.....	165
Написание программы с использованием АСД .....	167
Объекты АСД.....	169
Обход АСД .....	169
Расширение flake8 с помощью проверок АСД.....	171
Написание класса.....	172

Игнорирование нерелевантного кода .....	172
Проверка наличия правильного декоратора .....	173
Поиск self .....	174
Быстрое знакомство с Ну .....	176
Итоги.....	178
Пол Тальямонте об АСД и Ну.....	179
<b>Глава 10. Производительность и оптимизация .....</b>	<b>182</b>
Структуры данных .....	182
Понимание поведения кода через профилирование .....	185
cProfile.....	185
Дизассемблинг модулем dis.....	188
Эффективное объявление функций .....	190
Упорядоченные списки и bisect.....	191
Именованные кортежи и Slots.....	194
Мемоизация.....	200
Быстрый Python с PyPy .....	202
Zero-copy с протоколом буфера .....	203
Итоги.....	209
Виктор Стиннер об оптимизации .....	209
<b>Глава 11. Масштабирование и архитектура.....</b>	<b>212</b>
Многопоточность в Python и ее ограничения .....	212
Многопроцессность против многопоточности .....	214
Событийно-ориентированная архитектура.....	216
Другие опции и asyncio.....	218
Сервис-ориентированная архитектура .....	220
Межпроцессорное взаимодействие с ZeroMQ.....	221
Итоги.....	223
<b>Глава 12. Управление реляционными базами данных.....</b>	<b>224</b>
Использование RDBMS и ORM.....	224
Бэкенд баз данных .....	227

---

Потоковые данные с Flask и PostgreSQL.....	228
Создание приложения потоковых данных.....	228
Создание приложения .....	231
Димитри Фонтейн о базах данных .....	233
<b>Глава 13. Пишите меньше, программируйте больше .....</b>	<b>239</b>
Организация поддержки Python 2 и 3 с помощью six .....	239
Строки и Юникод.....	240
Обработка перемещения модулей .....	241
Модуль modernize.....	241
Использование Python как Lisp для одиночной диспетчеризации .....	242
Создание универсального метода в Lisp .....	242
Универсальные методы в Python .....	244
Контекстный менеджер .....	246
Меньше шаблонов с attr .....	250
Итоги.....	252