

Содержание

Об авторе	13
Введение	14
О чем эта книга	14
Соглашения, принятые в книге	16
Где найти код примеров	17
Ждем ваших отзывов!	18
Глава 1. Мыслим категориями языка Python	19
Рекомендация 1. Следите за тем, какую версию Python вы используете	19
Рекомендация 2. Руководствуйтесь правилами стилевого оформления программ, изложенными в документе PEP 8	21
Рекомендация 3. Знайте о различиях между типами строк <code>bytes</code> , <code>str</code> и <code>unicode</code>	24
Рекомендация 4. Заменяйте сложные выражения вспомогательными функциями	27
Рекомендация 5. Умейте работать со срезами последовательностей	30
Рекомендация 6. Избегайте совместного использования индексов начала, конца и шага в одном срезе	33
Рекомендация 7. Используйте генераторы списков вместо функций <code>map()</code> и <code>filter()</code>	36
Рекомендация 8. Избегайте использования более двух выражений в генераторах списков	37
Рекомендация 9. По возможности используйте выражения-генераторы вместо генераторов длинных списков	40
Рекомендация 10. По возможности используйте функцию <code>enumerate()</code> вместо функции <code>range()</code>	42

Рекомендация 11. Используйте функцию <code>zip()</code> для параллельной обработки итераторов	43
Рекомендация 12. Избегайте использования блоков <code>else</code> после циклов <code>for</code> и <code>while</code>	46
Рекомендация 13. Старайтесь использовать возможности каждого из блоков конструкции <code>try/except/else/finally</code>	49
Глава 2. Функции	53
Рекомендация 14. Использование исключений предпочтительнее возврата значения <code>None</code>	53
Рекомендация 15. Знайте, как замыкания взаимодействуют с областью видимости переменных	56
Рекомендация 16. Не упускайте возможность использовать генераторы вместо возврата списков	61
Рекомендация 17. Не забывайте о мерах предосторожности при итерировании аргументов	64
Рекомендация 18. Снижайте визуальный шум с помощью переменного количества позиционных аргументов	69
Рекомендация 19. Обеспечивайте опциональное поведение с помощью именованных аргументов	71
Рекомендация 20. Используйте значение <code>None</code> и средство <code>Docstrings</code> при задании динамических значений по умолчанию для аргументов	75
Рекомендация 21. Повышайте ясность кода, используя именованные аргументы	78
Глава 3. Классы и наследование	83
Рекомендация 22. Отдавайте предпочтение структуризации данных с помощью классов, а не словарей или кортежей	83
Рекомендация 23. Принимайте функции вместо классов в случае простых интерфейсов	89
Рекомендация 24. Используйте полиморфизм <code>@classmethod</code> для конструирования объектов обобщенным способом	94
Рекомендация 25. Инициализация родительских классов с помощью встроенной функции <code>super()</code>	99
Рекомендация 26. Используйте множественное наследование лишь для примесных вспомогательных классов	104
Рекомендация 27. Предпочитайте общедоступные атрибуты закрытым	109

8 Содержание

Рекомендация 28. Используйте наследование от классов из модуля <code>collections.abc</code> для создания пользовательских контейнерных типов	114
Глава 4. Метаклассы и атрибуты	119
Рекомендация 29. Используйте простые атрибуты вместо методов <code>get()</code> и <code>set()</code>	119
Рекомендация 30. Старайтесь использовать декораторы <code>@property</code> вместо рефакторинга атрибутов	124
Рекомендация 31. Используйте дескрипторы для повторно используемых методов <code>@property</code>	128
Рекомендация 32. Используйте методы <code>__getattr__()</code> , <code>__getattribute__()</code> и <code>__setattr__()</code> для отложенных атрибутов	134
Рекомендация 33. Верификация подклассов с помощью метаклассов	140
Рекомендация 34. Регистрируйте существование классов с помощью метаклассов	142
Рекомендация 35. Аннотирование атрибутов классов с помощью метаклассов	147
Глава 5. Одновременность и параллелизм	151
Рекомендация 36. Использование модуля <code>subprocess</code> для управления дочерними процессами	152
Рекомендация 37. Используйте потоки для блокирования операций ввода-вывода, но не для параллелизма	156
Рекомендация 38. Используйте класс <code>Lock</code> для предотвращения гонки данных в потоках	161
Рекомендация 39. Использование очередей для координации работы потоков	165
Рекомендация 40. Используйте сопрограммы для одновременного выполнения нескольких функций	173
Рекомендация 41. Старайтесь использовать модуль <code>concurrent.futures</code> для обеспечения истинного параллелизма	183
Глава 6. Встроенные модули	189
Рекомендация 42. Определяйте декораторы функций с помощью модуля <code>functools.wraps</code>	189
Рекомендация 43. Обеспечивайте возможность повторного использования блоков <code>try/finally</code> с помощью инструкций <code>contextlib</code> и <code>with</code>	192

Рекомендация 44. Повышайте надежность встроенного модуля <code>pickle</code> с помощью модуля <code>copyreg</code>	196
Рекомендация 45. Используйте модуль <code>datetime</code> вместо модуля <code>time</code> для локальных часов	202
Рекомендация 46. Используйте встроенные алгоритмы и структуры данных	206
Рекомендация 47. Используйте класс <code>Decimal</code> , когда на первый план выходит точность	211
Рекомендация 48. Знайте, где искать модули, разработанные сообществом Python	214
Глава 7. Совместная работа	217
Рекомендация 49. Снабжайте строками документирования каждую функцию, класс и модуль	217
Рекомендация 50. Используйте пакеты для организации модулей и предоставления стабильных API	222
Рекомендация 51. Изолируйте вызывающий код от API, определяя базовое исключение <code>Exception</code>	228
Рекомендация 52. Знайте, как устранять циклические зависимости	231
Рекомендация 53. Используйте виртуальные среды для изолированных и воспроизводимых зависимостей	237
Глава 8. Производство	245
Рекомендация 54. Используйте код с областью видимости модуля для конфигурирования сред развертывания	245
Рекомендация 55. Используйте строки <code>perg</code> для вывода отладочной информации	248
Рекомендация 56. Тестируйте любой код с помощью модуля <code>unittest</code>	251
Рекомендация 57. Используйте интерактивную отладку с помощью пакета <code>pdb</code>	255
Рекомендация 58. Сначала — профилирование, затем — оптимизация	257
Рекомендация 59. Используйте модуль <code>tracemalloc</code> для контроля памяти и предотвращения ее утечки	262
Предметный указатель	267