

*Эффективное объектно-ориентированное
программирование*

3-е Издание
охватывает Python 2.5

Изучаем

Python



O'REILLY®

Марк Лутц

Learning Python

Third Edition

Mark Lutz

O'REILLY®

Изучаем Python

Третье издание

Марк Лутц



Санкт-Петербург — Москва
2009

Марк Лутц
Изучаем Python, 3-е издание

Перевод А. Киселева

| | |
|----------------------|--------------------|
| Главный редактор | <i>А. Галунов</i> |
| Зав. редакцией | <i>Н. Макарова</i> |
| Выпускающий редактор | <i>П. Щеголев</i> |
| Научный редактор | <i>Б. Попов</i> |
| Редактор | <i>Ю. Бочина</i> |
| Корректор | <i>С. Минин</i> |
| Верстка | <i>Д. Орлова</i> |

Лутц М.

Изучаем Python, 3-е издание – Пер. с англ. – СПб.: Символ-Плюс, 2009. – 848 с., ил.

ISBN 978-5-93286-138-7

Мощный, переносимый и легкий в использовании язык программирования Python идеально подходит для разработки самостоятельных программ и сценариев. Эта книга позволит быстро и эффективно овладеть базовыми основами языка Python независимо от степени предварительного знакомства с ним.

Третье издание «Изучаем Python» основано на материалах учебных курсов, которые автор, Марк Лутц, ведет уже на протяжении десяти лет. В книге представлены основные типы объектов в языке Python, порядок их создания и работы с ними, а также функции как основной процедурный элемент языка. Рассматриваются методы работы с модулями и дополнительными объектно-ориентированными инструментами языка Python – классами. Включены описания моделей и инструкций обработки исключений, а также обзор инструментов разработки, используемых при создании крупных программ. Обсуждаются изменения в ожидаемой версии 3.0. В конце глав представлены упражнения и вопросы, которые позволят применить новые знания на практике и проверить, насколько хорошо усвоен материал.

Если вы хотите понять, почему выбирают Python такие компании, как Google и Intel, Cisco и Hewlett-Packard, почему этот язык используют в NASA для научных вычислений, то эта книга станет для вас лучшей отправной точкой.

ISBN 978-5-93286-138-7

ISBN 978-0-596-51398-6 (англ)

© Издательство Символ-Плюс, 2009

Authorized translation of the English edition © 2008 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Все права на данное издание защищены Законодательством РФ, включая право на полное или частичное воспроизведение в любой форме. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих фирм.

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7,
тел. (812) 324-5353, www.symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.

Подписано в печать 21.11.2008. Формат 70×100¹/16. Печать офсетная.

Объем 53 печ. л. Тираж 2000 экз. Заказ №

Отпечатано с готовых диапозитивов в ГУП «Типография «Наука»
199034, Санкт-Петербург, 9 линия, 12.

Посвящается Вере.

Ты жизнь моя.

Оглавление

| | |
|---|-----------|
| Предисловие | 15 |
| I. Введение | 33 |
| 1. Python в вопросах и ответах | 35 |
| Почему программисты используют Python? | 35 |
| Является ли Python «языком сценариев»? | 38 |
| Все хорошо, но есть ли у него недостатки? | 40 |
| Кто в наше время использует Python? | 41 |
| Что можно делать с помощью Python? | 42 |
| В чем сильные стороны Python? | 46 |
| Какими преимуществами обладает Python перед языком X? | 54 |
| В заключение | 55 |
| Закрепление пройденного | 55 |
| 2. Как Python запускает программы | 58 |
| Введение в интерпретатор Python | 58 |
| Выполнение программы | 60 |
| Разновидности модели выполнения | 65 |
| В заключение | 70 |
| Закрепление пройденного | 71 |
| 3. Как пользователь запускает программы | 72 |
| Интерактивный режим | 72 |
| Системная командная строка и файлы | 77 |
| Щелчок на ярлыке файла | 83 |
| Импортирование и перезагрузка модулей | 86 |
| Пользовательский интерфейс IDLE | 92 |
| Другие интегрированные среды разработки | 98 |
| Встраивание вызовов | 100 |
| Фиксированные исполняемые двоичные файлы | 101 |
| Возможность запуска программ из текстового редактора | 101 |
| Прочие возможности запуска | 101 |
| Будущие возможности | 102 |

| | |
|---|------------|
| Какие способы следует использовать? | 102 |
| В заключение | 103 |
| Закрепление пройденного | 103 |
| II. Типы и операции | 109 |
| 4. Введение в типы объектов языка Python | 111 |
| Зачем нужны встроенные типы? | 112 |
| Числа | 115 |
| Строки | 116 |
| Списки | 123 |
| Словари | 127 |
| Кортежи | 134 |
| Файлы | 135 |
| Другие базовые типы | 136 |
| В заключение | 139 |
| Закрепление пройденного | 140 |
| 5. Числа | 142 |
| Числовые типы в Python | 142 |
| Операторы выражений | 145 |
| Числа в действии | 150 |
| Другие числовые типы | 160 |
| В заключение | 164 |
| Закрепление пройденного | 164 |
| 6. Интерлюдия о динамической типизации | 166 |
| Отсутствие инструкций объявления | 166 |
| Разделяемые ссылки | 171 |
| Динамическая типизация повсюду | 176 |
| В заключение | 177 |
| Закрепление пройденного | 177 |
| 7. Строки | 179 |
| Литералы строк | 181 |
| Строки в действии | 190 |
| Форматирование строки | 200 |
| Строковые методы | 204 |
| Общие категории типов | 211 |
| В заключение | 212 |
| Закрепление пройденного | 213 |
| 8. Списки и словари | 215 |
| Списки | 215 |
| Списки в действии | 218 |

| | |
|---|------------|
| Словари | 224 |
| Словари в действии | 227 |
| В заключение | 237 |
| Закрепление пройденного | 237 |
| 9. Кортежи, файлы и все остальное | 239 |
| Кортежи | 240 |
| Кортежи в действии | 241 |
| Файлы | 244 |
| Пересмотренный перечень категорий типов | 252 |
| Гибкость объектов | 253 |
| Сравнения, равенство и истина | 257 |
| Иерархии типов данных в языке Python | 260 |
| Другие типы в Python | 262 |
| Ловушки встроенных типов | 262 |
| В заключение | 265 |
| Закрепление пройденного | 265 |
| III. Инструкции и синтаксис | 271 |
| 10. Введение в инструкции языка Python | 273 |
| Структура программы на языке Python | 273 |
| История о двух if | 275 |
| Короткий пример: интерактивные циклы | 284 |
| В заключение | 289 |
| Закрепление пройденного | 290 |
| 11. Присваивание, выражения и print | 292 |
| Инструкции присваивания | 292 |
| Инструкции выражений | 305 |
| Инструкция print | 307 |
| В заключение | 312 |
| Закрепление пройденного | 313 |
| 12. Условная инструкция if | 315 |
| Условные инструкции if | 315 |
| Синтаксические правила языка Python | 318 |
| Проверка истинности | 323 |
| В заключение | 328 |
| Закрепление пройденного | 328 |
| 13. Циклы while и for | 330 |
| Циклы while | 330 |
| break, continue, pass и else | 332 |

| | |
|--|------------|
| Циклы <code>for</code> | 337 |
| Итераторы: первое знакомство | 342 |
| Приемы программирования циклов | 349 |
| Генераторы списков: первое знакомство | 357 |
| В заключение | 361 |
| Закрепление пройденного | 362 |
| 14. Документация | 364 |
| Источники документации в языке Python | 364 |
| Типичные ошибки программирования | 377 |
| В заключение | 379 |
| Закрепление пройденного | 380 |
| IV. Функции | 383 |
| 15. Основы функций | 385 |
| Зачем нужны функции? | 386 |
| Создание функций | 387 |
| Первый пример: определения и вызовы | 390 |
| Второй пример: пересечение последовательностей | 393 |
| В заключение | 396 |
| Закрепление пройденного | 396 |
| 16. Области видимости и аргументы | 398 |
| Правила видимости | 398 |
| Инструкция <code>global</code> | 406 |
| Области видимости и вложенные функции | 410 |
| Передача аргументов | 418 |
| Специальные режимы сопоставления аргументов | 423 |
| В заключение | 436 |
| Закрепление пройденного | 437 |
| 17. Расширенные возможности функций | 440 |
| Анонимные функции: <code>lambda</code> | 440 |
| Применение функций к аргументам | 447 |
| Отображение функций на последовательности: <code>map</code> | 449 |
| Средства функционального программирования: <code>filter</code> и <code>reduce</code> | 451 |
| Еще раз о генераторах списков: отображения | 452 |
| Еще раз об итераторах: генераторы | 459 |
| Хронометраж итерационных альтернатив | 466 |
| Концепции проектирования функций | 469 |
| Типичные ошибки при работе с функциями | 472 |
| В заключение | 477 |
| Закрепление пройденного | 477 |

| | |
|---|-----|
| V. Модули | 483 |
| 18. Модули: общая картина | 485 |
| Зачем нужны модули? | 486 |
| Архитектура программы на языке Python | 487 |
| Как работает импорт | 490 |
| В заключение | 498 |
| Закрепление пройденного | 499 |
| 19. Основы программирования модулей | 501 |
| Создание модуля | 501 |
| Использование модулей | 502 |
| Пространства имен модулей | 509 |
| Повторная загрузка модулей | 514 |
| В заключение | 518 |
| Закрепление пройденного | 519 |
| 20. Пакеты модулей | 521 |
| Основы операции импортирования пакетов | 521 |
| Пример импортирования пакета | 525 |
| Зачем используется операция импортирования пакетов? | 527 |
| В заключение | 531 |
| Закрепление пройденного | 532 |
| 21. Дополнительные возможности модулей | 533 |
| Скрытие данных в модулях | 533 |
| Включение будущих возможностей языка | 534 |
| Смешанные режимы использования: <code>__name__</code> и <code>__main__</code> | 535 |
| Изменение пути поиска модулей | 538 |
| Расширение <code>import as</code> | 539 |
| Синтаксис относительного импорта | 539 |
| Концепции проектирования модулей | 543 |
| Типичные проблемы при работе с модулями | 547 |
| В заключение | 555 |
| Закрепление пройденного | 555 |
| VI. Классы и ООП | 559 |
| 22. ООП: общая картина | 561 |
| Зачем нужны классы? | 562 |
| ООП с высоты 30 000 футов | 564 |
| В заключение | 575 |
| Закрепление пройденного | 575 |

| | |
|--|-----|
| 23. Основы программирования классов | 577 |
| Классы генерируют множество экземпляров объектов | 577 |
| Классы адаптируются посредством наследования | 582 |
| Классы могут переопределять операторы языка Python | 586 |
| Самый простой в мире класс на языке Python | 589 |
| В заключение | 592 |
| Закрепление пройденного | 592 |
| 24. Подробнее о программировании классов | 595 |
| Инструкция class | 595 |
| Методы | 599 |
| Наследование | 602 |
| Перегрузка операторов | 606 |
| Пространства имен: окончание истории | 624 |
| Более реалистичный пример | 631 |
| В заключение | 635 |
| Закрепление пройденного | 635 |
| 25. Шаблоны проектирования с классами | 637 |
| Python и ООП | 637 |
| Классы как записи | 639 |
| ООП и наследование: взаимосвязи типа «является» | 641 |
| ООП и композиция: взаимосвязи типа «имеет» | 643 |
| ООП и делегирование | 648 |
| Множественное наследование | 649 |
| Классы – это объекты: фабрики универсальных объектов | 653 |
| Методы – это объекты: связанные и несвязанные методы | 655 |
| Еще раз о строках документирования | 657 |
| Классы и модули | 659 |
| В заключение | 659 |
| Закрепление пройденного | 660 |
| 26. Дополнительные возможности классов | 661 |
| Расширение встроенных типов | 661 |
| Псевдочастные атрибуты класса | 665 |
| Классы нового стиля | 668 |
| Статические методы и методы класса | 677 |
| Декораторы функций | 681 |
| Типичные проблемы при работе с классами | 684 |
| В заключение | 690 |
| Закрепление пройденного | 690 |

| | |
|---|-----|
| VII. Исключения и инструменты | 699 |
| 27. Основы исключений | 701 |
| Зачем нужны исключения? | 702 |
| Обработка исключений: краткий обзор | 704 |
| Инструкция try/except/else | 709 |
| Инструкция try/finally | 716 |
| Объединенная инструкция try/except/finally | 718 |
| Инструкция raise | 722 |
| Инструкция assert | 725 |
| Контекстные менеджеры with/as | 726 |
| В заключение | 731 |
| Закрепление пройденного | 732 |
| 28. Объекты исключений | 733 |
| Исключения на основе строк | 734 |
| Исключения на основе классов | 735 |
| Общие формы инструкции raise | 747 |
| В заключение | 748 |
| Закрепление пройденного | 749 |
| 29. Использование исключений | 751 |
| Вложенные обработчики исключений | 751 |
| Идиомы исключений | 756 |
| Советы по применению исключений | 760 |
| Типичные проблемы при работе с исключениями | 764 |
| Заклучение по основам языка | 766 |
| В заключение | 771 |
| Закрепление пройденного | 772 |
| VIII. Приложения | 775 |
| A. Установка и настройка | 777 |
| B. Решения упражнений | 786 |
| Алфавитный указатель | 824 |

Предисловие

Эта книга представляет собой введение в язык программирования Python. Python – это популярный язык программирования, используемый как для разработки самостоятельных программ, так и для создания прикладных сценариев в самых разных областях применения. Это мощный, переносимый, простой в использовании и свободно распространяемый язык программирования.

Цель этой книги – позволить вам быстро овладеть основными принципами базового языка Python независимо от уровня вашей подготовки. Прочитав эту книгу, вы получите объем знаний, достаточный для использования Python в самых разных областях.

О третьем издании

За четыре года, прошедшие с момента выхода второго издания книги в 2003 году, произошли существенные изменения как в самом языке Python, так и в темах, которые я представляю на своих курсах обучения языку Python. Я постарался оставить как можно больше из предыдущего издания, кроме того, это новое издание отражает изменения последнего времени, появившиеся в самом языке Python и в обучении языку Python. Помимо этого была несколько изменена структура книги.

Изменения в языке Python

Если говорить о версии языка, это издание описывает Python 2.5 и отражает все изменения, появившиеся в языке с момента выхода второго издания книги. (Во втором издании описывался язык Python 2.2 и некоторые нововведения версии 2.3.) Кроме того, в этом издании обсуждаются изменения, которые ожидаются в версии Python 3.0. Ниже приводится список основных тем, касающихся языка программирования, которые вы найдете в этом издании:

- Новая условная конструкция `B if A else C` (глава 12).
- Оператор контекста `with/as` (глава 27).
- Унификация конструкции `try/except/finally` (глава 27).
- Синтаксис относительного импорта (глава 21).
- Выражения-генераторы (глава 17).
- Новые особенности функций-генераторов (глава 17).

- **Функции-декораторы** (глава 26).
- **Объектный тип множества** (глава 5).
- **Новые встроенные функции:** `sorted`, `sum`, `any`, `all`, `enumerate` (главы 4 и 13).
- **Объектный тип десятичных чисел с фиксированной точностью представления** (глава 5).
- **Расширенные сведения о файлах, списках, итераторах и многом другом** (главы 13 и 17).
- **Новые инструменты разработки:** `Eclipse`, `dustutils`, `unittest` и `doctest`, расширения `_IDLE`, `Shedskin` и т. д. (главы 3 и 29).

Менее значительные изменения в языке (такие, как широко используемые значения `True` и `False`, новая функция `sys.exc_info`, которая возвращает подробную информацию об исключении, и обработка строковых исключений, методы для работы со строками и встроенные функции `apply` и `reduce`) обсуждаются на протяжении всей книги. Кроме того, здесь приводится расширенное описание некоторых новых особенностей, впервые появившихся в предыдущем издании, включая третье измерение при работе со срезами и передачу произвольного числа аргументов функции, включая функцию `apply`.

Изменения в обучении языку Python

Кроме уже перечисленных изменений в самом языке это издание книги дополнено новыми темами и примерами, наработанными мною при преподавании на курсах обучения языку Python в последние годы. Например, здесь вы найдете:

- Новую главу о встроенных типах (глава 4).
- Новую главу о синтаксических конструкциях (глава 10).
- Полностью новую главу о динамической типизации с углубленным освещением этого вопроса (глава 6).
- Расширенное введение в ООП (глава 22).
- Новые примеры работы с файлами, областями видимости, вложенными инструкциями, классами, исключениями и т. д.

Множество изменений и дополнений было сделано, чтобы облегчить чтение книги начинающим программистам. Учитывая опыт преподавания языка Python на курсах, обсуждение некоторых моментов было перенесено в другие, более соответствующие тематике разделы. Например, описание списков и итераторов теперь приводится вместе с описанием оператора цикла `for`, а не с описанием функциональных инструментов, как это было ранее.

Поскольку книга фактически превратилась в стандартный учебник по Python, в этом издании существенно расширено описание основ языка, изложение материала стало более полным и наглядным.

В дополнение в этом издании приводится полностью обновленный комплект советов и рекомендаций, подобранных из опыта преподавания в течение последних 10 лет и практического использования Python в течение последних 15 лет. Также были дополнены и расширены учебные упражнения – с целью отразить наиболее удачные современные приемы программирования на языке Python, его новые особенности и показать наиболее типичные ошибки, которые совершают начинающие программисты на моих курсах. Вообще основы языка в этом издании обсуждаются более широко, чем в предыдущих изданиях, во-первых, потому что сам язык был расширен, а во-вторых, потому что я добавил немало информации, которая имеет важное значение для практического применения.

Структурные изменения в этом издании

Учитывая разросшийся объем книги, материал, как и в предыдущем издании, разбит на несколько частей. Чтобы облегчить усвоение основ языка, весь материал поделен на несколько частей, каждая из которых содержит несколько глав. Например, типы и инструкции теперь описываются в двух разных частях, в каждой из которых основным типам и инструкциям отведены отдельные главы. Новая структура позволяет сказать больше, не перегружая читателя при этом. При переработке материала упражнения и описания наиболее распространенных ошибок были перемещены из конца главы в конец части, и теперь они приводятся в конце последней главы каждой части.

Упражнения, которые приводятся в конце каждой части, я также дополнил в этом издании кратким обзором и контрольными вопросами в конце каждой главы, что должно помочь вам «пролистать» прочитанный материал. Каждая глава заканчивается серией вопросов, которые помогут вам проверить, насколько хорошо вы поняли изложенный материал. В отличие от упражнений в конце каждой части, решения для которых приводятся в приложении В, ответы на вопросы в конце каждой главы следуют непосредственно за вопросами. Я рекомендую просматривать ответы, даже если вы уверены, что правильно ответили на вопросы, потому что эти ответы, кроме всего, являются кратким обзором только что пройденной темы.

Несмотря на наличие новых тем, эта книга по-прежнему ориентирована на тех, кто только начинает знакомство с языком Python. Она задумывалась как учебник по Python для программистов¹. Во многом она унаследовала из первых двух изданий и материал, и структуру, и акценты подачи материала. Где это необходимо, я расширил вводную

¹ Под «программистами» я подразумеваю всех, кто в прошлом написал хотя бы одну строчку программного кода на любом языке программирования. Если вы не относитесь к этой категории, эта книга все равно будет вам полезна, но вы должны знать, что она в основном обучает языку Python, а не основам программирования.

часть для начинающих и отделил более сложные темы от основного потока обсуждения, чтобы не усложнять описание основ. Поскольку это издание в значительной степени основано на проверенном временем опыте преподавания, оно, как и первые два, может служить вводным курсом для самостоятельного изучения языка Python.

Ограничение области применения книги

Третье издание представляет собой учебник по основам языка программирования Python и ничего больше. Здесь приводятся всесторонние сведения о языке, которые необходимо знать, прежде чем приступить к практическому его использованию. Материал подается в порядке постепенного усложнения и дает полное представление о языке программирования, не фокусируясь на областях его применения.

Для некоторых «изучить Python» означает потратить час-другой на изучение руководств в сети Интернет. Такой подход пригоден для опытных программистов – в конце концов, Python – довольно простой язык по сравнению с другими языками программирования. Однако проблема такого ускоренного изучения состоит в том, что на практике программисты часто сталкиваются с необычными случаями необъяснимого изменения значений переменных, параметров по умолчанию и т. д. Цель этой книги состоит в том, чтобы дать твердое понимание основ языка Python, чтобы даже самые необычные случаи находили свое объяснение.

Это ограничение было введено преднамеренно. Ограничившись обсуждением основ языка, мы можем заняться более глубоким и полным их исследованием. Более полное обсуждение темы прикладного использования Python и справочные материалы, не вошедшие в эту книгу, вы найдете в других публикациях издательства O'Reilly, таких как «Programming Python», «Python Cookbook», «Python in a Nutshell» и «Python Pocket Reference». Цель книги, которую вы сейчас читаете, состоит исключительно в изучении языка Python, чтобы потом вы смогли применять его для решения прикладных задач, независимо от предметной области, в которой вы работаете.

По этой причине некоторые справочные материалы и разделы предыдущих изданий (примерно 15 процентов предыдущего издания) были вырезаны с целью расширить разделы с описанием основ языка Python. Благодаря этому читатели этого издания получают более полное представление об основах языка программирования и – как следствие – более полезный учебник по языку Python. В качестве заключительного упражнения в конце книги (глава 29) приводятся несколько усложненных примеров для самостоятельного изучения.

Об этой книге

В этом разделе приводятся некоторые наиболее важные замечания об этой книге в целом, не зависящие от номера издания. Никакая книга

не способна удовлетворить все нужды и потребности читателя, поэтому важно понимать основные цели книги.

Предварительные условия

В действительности книга не предъявляет никаких предварительных условий. Она с успехом использовалась как начинающими программистами, так и умудренными опытом ветеранами. Если у вас есть желание изучать Python, эта книга наверняка поможет вам. Наличие у читателя некоторого опыта в программировании не является обязательным, но будет совсем не лишним.

Эта книга задумывалась как введение в Python для программистов. Возможно, она не идеальна для тех, кто раньше никогда не имел дела с компьютерами (например, мы не будем тратить время, чтобы объяснять, что такое компьютер), но я не делал никаких предположений о наличии у читателя опыта программирования или об уровне его подготовки.

С другой стороны, – я не считаю нужным обижать читателей, предполагая, что они «чайники», что бы это ни означало, – писать полезные программы на языке Python просто, и эта книга покажет, как это делается. В книге Python иногда противопоставляется другим языкам программирования, таким как C, C++, Java™ и Pascal, но эти сравнения можно просто игнорировать, если ранее вам не приходилось работать с этими языками программирования.

Отношения этой книги с другими книгами

Эта книга охватывает все основные аспекты языка программирования Python, но при этом я старался ограничить круг обсуждаемых тем, чтобы уменьшить объем книги. Для сохранения простоты в ней рассматриваются самые основные понятия, используются небольшие и очевидные примеры и опущены некоторые незначительные детали, которые вы найдете в справочных руководствах. По этой причине данная книга скорее должна рассматриваться как введение, как первый шаг к другим, более специализированным и более полным книгам.

Например, мы не будем говорить об интеграции Python/C – это слишком сложная тема, которая, однако, является центральной для многих систем, основанных на применении Python. Мы также не будем говорить об истории развития Python и о процессе его разработки. А таких популярных применений Python, как создание графического интерфейса, разработка системных инструментов и работа с сетью, мы коснемся лишь очень кратко, если они вообще будут упоминаться. Естественно, при таком подходе из поля зрения выпадает значительная часть общей картины.

Вообще говоря, Python стоит на более высоком качественном уровне относительно других языков в мире языков сценариев. Некоторые из его идей требуют более глубокого изучения, чем может вместить эта книга, поэтому с моей стороны было бы ответственно порекомендовать

продолжить его изучение после того, как вы закончите читать эту книгу. Я надеюсь, что большинство читателей продолжит изучение принципов разработки приложений на этом языке, обратившись к другим источникам информации.

Ориентированная в основном на начинающих программистов, книга «Изучаем Python» может быть дополнена другими книгами издательства O'Reilly о языке Python. Например, существует еще одна моя книга «Programming Python», содержащая более объемные и полные примеры наряду с описанием приемов прикладного программирования, которая задумывалась как продолжение книги, которую вы сейчас читаете. Текущие издания книг «Изучаем Python» и «Programming Python» представляют собой две части курса обучения, который преподает автор, – основы языка и прикладное программирование. Кроме того, в качестве справочника можно использовать еще одну книгу издательства O'Reilly, «Python Pocket Reference», где приводятся некоторые подробности, опущенные здесь.

Для дальнейшего изучения можно порекомендовать книги, содержащие дополнительные сведения, примеры или особенности использования языка Python в определенных прикладных областях, таких как веб-приложения и создание графических интерфейсов. Например, книги «Python in a Nutshell» (O'Reilly) и «Python Essential Reference» (Sams) содержат справочную информацию. Книга «Python Cookbook» (O'Reilly) представляет собой сборник примеров для тех, кто уже знаком с приемами прикладного программирования. Поскольку выбор книг является делом достаточно субъективным, я рекомендую вам самостоятельно поискать такие, которые наиболее полно будут отвечать вашим потребностям. Неважно, какие книги вы выберете, главное чтобы вы помнили, что для дальнейшего изучения Python вам необходимы более реалистичные примеры, чем приводятся здесь.

На мой взгляд, эта книга будет для вас отличным учебником начального уровня, даже несмотря на ее ограниченность (и скорее всего именно поэтому). Здесь вы найдете все, что необходимо знать, прежде чем приступать к созданию программ и сценариев на языке Python. К тому моменту, когда вы закончите чтение этой книги, вы изучите не только сам язык, но и начнете понимать, как лучше применить его к решению ваших повседневных задач. Кроме того, у вас будет все необходимое для изучения более сложных тем и примеров, которые будут встречаться на вашем пути.

Стиль и структура книги

Эта книга основана на материалах трехдневных практических курсов изучения языка Python. В конце каждой главы содержится список контрольных вопросов, а в конце последней главы каждой части – упражнения. Ответы на контрольные вопросы приводятся непосредственно в самих главах, а примеры решения упражнений – в приложении В.

Контрольные вопросы подобраны так, что они представляют собой краткий обзор рассмотренного материала, а упражнения спроектированы так, чтобы сразу же научить вас правильному стилю программирования и, как правило, каждое упражнение соответствует одному из ключевых аспектов курса.

Я настоятельно рекомендую прорабатывать контрольные вопросы и упражнения в ходе чтения книги не только для того, чтобы получить опыт программирования на Python, но и потому, что в упражнениях поднимаются проблемы, которые не обсуждаются нигде в книге. Ответы на вопросы в главах и примеры решения упражнений в приложении В в случае необходимости помогут вам выйти из затруднительных положений (вы можете заглядывать в ответы так часто, как это потребуетя).

Общая структура книги также следует структуре учебного курса. Так как эта книга задумывалась как быстрое введение в основы языка программирования, изложение материала организовано так, чтобы оно отражало основные особенности языка, а не частности. Мы будем двигаться от простого к сложному: от встроенных типов объектов к инструкциям, элементам программ и т. д. Каждая глава является полным и самостоятельным описанием одной темы, но каждая последующая глава основана на идеях, введенных в предыдущих главах (например, когда речь пойдет о классах, я буду исходить из предположения, что вы уже знаете, как создаются функции), поэтому для большинства читателей имеет смысл читать книгу последовательно.

В общих чертах, эта книга описывает язык программирования Python при движении от простого к сложному. Каждая часть посвящена отдельной крупной характеристике языка – типам, функциям и т. д. В большинстве своем примеры являются законченными небольшими сценариями (некоторые из них являются достаточно искусственными, но они иллюстрируют достижение поставленной цели). Если быть более точным, здесь вы найдете:

Часть I. *Введение*

Изучение Python мы начнем с общего обзора этого языка и с ответов на очевидно возникающие вопросы: почему кто-то использует этот язык, для решения каких задач он может использоваться и т. д. В первой главе рассматриваются основные идеи, лежащие в основе технологии, которые должны дать вам некоторые начальные представления. Далее начинается сугубо технический материал книги. Здесь мы рассмотрим, как выполняют программы человек и интерпретатор Python. Цель этой части книги состоит в том, чтобы дать вам начальные сведения, которые позволят вам работать с последующими примерами и упражнениями.

Часть II. *Типы и операции*

Далее мы приступим к исследованию языка программирования Python и начнем его изучение с основных типов объектов, таких как числа, списки, словари и т. д. Обладая только этими инструментами,

вы уже сможете писать достаточно сложные программы. Это самая важная часть книги, потому что она закладывает основу для последующих глав. В этой части мы также рассмотрим динамическую типизацию и ссылки – ключевые аспекты языка Python.

Часть III. *Инструкции и синтаксис*

В следующей части вводятся *инструкции* языка Python – программный код на языке Python, который создает и обслуживает объекты. Здесь также будет представлена общая синтаксическая модель Python. Хотя эта часть в основном сосредоточена на описании синтаксиса, тем не менее, здесь приводятся сведения о дополнительных инструментальных средствах, таких как система PyDoc, и рассматриваются альтернативные стили написания программного кода.

Часть IV. *Функции*

В этой части мы начнем рассматривать высокоуровневые способы структурирования программ на языке Python. Функции предоставляют простой способ упаковки программного кода многократного использования и предотвращения появления избыточного кода. В этой части мы исследуем правила видимости программных элементов в языке Python, приемы передачи аргументов и многое другое.

Часть V. *Модули*

Модули Python позволяют организовать наборы инструкций и функций в виде крупных компонентов, и в этой части будет показано, как создавать модули, как их использовать и перезагружать. Здесь мы также рассмотрим некоторые более сложные темы, такие как пакеты модулей, перезагрузка модулей и переменная `__name__`.

Часть VI. *Классы и ООП*

Здесь мы приступим к исследованию объектно-ориентированного программирования (ООП). *Классы* – это необязательный, но очень мощный инструмент структурирования программного кода многократного использования. Здесь вы увидите, что классы по большей части используют идеи, которые будут описаны к этому моменту, а ООП в языке Python в основном представляет собой поиск имен в связанных объектах. Здесь вы также увидите, что объектно-ориентированный стиль программирования в языке Python не является обязательным, но может существенно сократить время разработки, особенно если речь идет о долгосрочных проектах.

Часть VII. *Исключения и инструменты*

Изучение языка мы закончим рассмотрением модели обработки исключительных ситуаций, а также кратким обзором инструментальных средств разработки, которые особенно удобны при разработке крупных программ (например, инструменты отладки и тестирования). Эта часть появляется в последний раз, в следующих версиях все исключения должны быть классами.

Часть VIII. Приложения

Книга заканчивается двумя приложениями, где приводятся рекомендации по использованию языка Python на различных платформах (Приложение А) и варианты решения упражнений, которые приводятся в конце каждой части (Приложение В). Ответы на контрольные вопросы, которые приводятся в конце каждой главы, находятся непосредственно в самих главах.

Обратите внимание: предметный указатель и оглавление могут использоваться для поиска информации, но в этой книге нет приложений со справочными материалами (эта книга – учебник, а не справочник). Как уже говорилось выше, в качестве справочников по синтаксису и встроенным особенностям языка Python вы можете использовать книгу «Python Pocket Reference» (O’Reilly) и справочники на сайте <http://www.python.org>.

Обновления книги

Книга продолжает улучшаться (и исправляются описанные ошибки и опечатки). Обновления, дополнения и исправления к этой книге можно найти в сети Интернет на одном из следующих сайтов:

<http://www.oreilly.com/catalog/9780596513986/>
(веб-страница книги на сайте издательства O’Reilly)

<http://www.rmi.net/~lutz> (сайт автора книги)

<http://www.rmi.net/~lutz/about-lp.html>
(веб-страница книги на сайте автора)

Последний из этих трех URL указывает на веб-страницу, где я выкладываю обновления, однако если эта ссылка окажется ошибочной, вам придется воспользоваться поисковой системой, чтобы восстановить ее. Если бы я был ясновидящим, я указал бы точную ссылку, но Интернет меняется быстрее, чем печатаются книги.

О программах в этой книге

Эта книга и все примеры программ в ней основаны на использовании Python 2.5. Я не стремлюсь предсказывать будущее, тем не менее, в ходе изучения мы будем обсуждать некоторые идеи, которые, как ожидается, будут реализованы в версии 3.0.

Однако, так как эта книга описывает базовые основы языка, можно быть уверенным, что большая часть из того, о чем здесь рассказывается, в следующих версиях Python изменится не очень сильно. Большая часть информации из этой книги применима и к более ранним версиям Python, кроме некоторых случаев; и, естественно, в случае использования расширений, которые появятся после выхода этой книги, ничего гарантировать нельзя.

Существует эмпирическое правило: лучшей версией Python является последняя его версия. Так как эта книга описывает основы языка, большинство сведений применимо к Jython – реализации Python на языке Java, а также к другим реализациям, описанным в главе 2.

Исходные тексты примеров, а также ответы к заданиям можно получить на веб-сайте книги по адресу <http://www.oreilly.com/catalog/9780596513986/>. Вас волнует вопрос, как запускать примеры? Он во всех подробностях обсуждается в главе 3, поэтому потерпите до этой главы.

В преддверии выхода Python 3.0

Первая альфа-версия Python 3.0 вышла как раз перед тем, как эта книга была отправлена в печать, уже после того, как она была написана. Официально это издание книги основано на линейке версий Python 2.x (в частности, на версии 2.5), но она была дополнена многочисленными примечаниями об ожидаемых изменениях в версии Python 3.0.

Официальная версия 3.0 выйдет как минимум через год после выхода этой книги и едва ли будет широко использоваться, по меньшей мере, еще два года. Однако, если вы приобрели эту книгу, когда версия 3.0 уже получила широкое распространение, в этом разделе вы найдете краткое описание изменений в языке, которое поможет вам выполнить переход на новую версию.

За редкими исключениями, в основном язык Python 3.0 будет соответствовать описываемому в книге и влияние этих изменений на типичный программный код будет весьма незначительным. То есть основы языка Python, о которых говорится в этой книге, не будут изменяться от версии к версии, и благодаря этому читатели могут с пользой для себя изучать эти основы, прежде чем переходить к рассмотрению особенностей, характерных для конкретных версий.

Однако, чтобы помочь вам в будущем, ниже приводится список основных отличий Python 3.0. Здесь же приводятся ссылки на главы, где обсуждаются или упоминаются эти изменения. Этот список упорядочен по возрастанию номеров глав. Некоторые из этих изменений могут быть реализованы в современной версии Python 2.5, а некоторые нет. Поскольку в данный момент для большинства читателей изучать этот список не имеет большого смысла, я рекомендую сначала прочитать эту книгу, чтобы изучить основы языка Python, а затем вернуться сюда позднее и ознакомиться с грядущими изменениями. В версии Python 3.0:

- Удалена нынешняя встроенная функция `execfile()`. Вместо нее следует использовать функцию `exec()` (глава 3).
- Встроенная функция `reload()` возможно будет удалена. Альтернатива пока не известна (главы 3 и 19).
- Преобразование в строковое представление с помощью обратных кавычек ``X`` будет недоступно: используйте функцию `repr(X)` (глава 5).

- Избыточная операция проверки на неравенство $X <> Y$ будет удалена: используйте операцию $X != Y$ (глава 5).
- Множества можно будет создавать с использованием синтаксиса литералов `{1, 3, 2}`, что эквивалентно используемой в настоящее время форме записи: `set([1, 3, 2])` (глава 5).
- Множества могут определяться программно: `{f(x) for x in S if P(x)}`, что эквивалентно используемой в настоящее время форме записи выражения-генератора: `set(f(x) for x in S if P(x))` (глава 5).
- Операция деления X/Y всегда возвращает число с плавающей точкой, даже если оба операнда являются целыми числами. Чтобы получить нынешнюю операцию деления с усечением дробной части, следует использовать $X//Y$ (глава 5).
- Существует единственный целочисленный тип `int`, который обеспечивает точность представления целых чисел, соответствующую нынешнему типу `long` (глава 5).
- Восьмеричные и двоичные литералы: текущая форма записи восьмеричных чисел `0666` будет вызывать ошибку: используйте вместо нее запись в форме `0o666`, соответствующим образом будет изменен и результат, возвращаемый функцией `oct()`. Также запись в форме `0b1010` будет эквивалентна числу 10, а функция `bin(10)` будет возвращать `"0b1010"` (глава 5).
- Строковый тип `str` поддерживает текст Unicode, а для представления строк с однобайтовыми символами создан новый тип `bytes` (например, для случаев, когда текст загружается из файлов в двоичном режиме). Тип `bytes` – это последовательность переменной длины малых целых чисел с интерфейсом, несколько отличающимся от `str` (глава 7).
- Появился новый дополнительный способ форматирования строк, так `"See {0}, {1} and {foo}".format("A", "B", foo="C")` вернет результат `"See A, B and C"` (глава 7).
- Метод словаря `D.has_key(X)` будет удален. Вместо него следует использовать проверку на членство `X in D` (главы 4 и 8).
- Сравнение (при сортировке) смешанных нечисловых типов вместо использования текущей реализации механизма упорядочения будет вызывать исключение (главы 8 и 9).
- Методы словаря `.keys()`, `.items()` и `.values()` вместо списков будут возвращать «представления» объектов, поддерживающие возможность выполнения итераций. Чтобы вернуться к прежней логике выполнения, следует выполнять принудительное преобразование с помощью `list()` (глава 8).
- Согласно предыдущему пункту следующий шаблон программирования будет неработоспособен: `k = D.keys(); k.sort();` вместо него следует использовать `k = sorted(D)` (главы 4 и 8).
- Встроенная функция `file()` может быть удалена. Вместо нее следует использовать функцию `open()` (глава 9).

- Встроенная функция `raw_input()` будет переименована в `input()`. Чтобы обеспечить поддержку логики работы нынешней функции `input()`, следует использовать `eval(input())` (глава 10).
- Инструкция выполнения строки программного кода `exec` снова станет встроенной функцией (глава 10).
- Появятся новые зарезервированные слова `as`, `with` и `nonlocal`. Согласно предыдущему пункту `exec` перестанет быть зарезервированным словом (глава 11).
- Инструкция печати станет функцией, поддерживающей большое число возможностей. Вместо `print x, y` следует использовать `print(x, y)`, а также можно использовать новые ключевые аргументы функции для настройки параметров печати: `file=sys.stdout`, `sep=" "` и `end="\n"` (глава 11).
- Появится расширенная реализация операции распаковки: инструкция, поддерживающая универсальную форму присваивания последовательности, такая как `a, b, *rest = some_sequence`, теперь будет работать, как и `*rest, a = stuff`. Таким образом, число элементов слева и справа от инструкции присваивания больше не должно совпадать (глава 11).
- Автоматический режим распаковки кортежей через присваивание последовательности для функций будет недоступен. Больше нельзя будет записать `def foo(a, (b, c)):`, вместо этого необходимо будет выполнять явное присваивание последовательности: `def foo(a, bc): b, c = bc` (глава 11).
- Встроенная функция `xrange()` будет переименована в `range()`. То есть в этой версии Python будет существовать только функция `range()` (глава 13).
- В протоколе итераций метод `X.next()` будет переименован в `X.__next__()`, и появится новая встроенная функция `next(X)`, которая будет вызывать метод `X.__next__()` объекта (главы 13 и 17).
- Встроенные функции `zip()`, `map()` и `filter()` будут возвращать итераторы. Чтобы вернуться к логике использования списков, следует применять `list()` (главы 13 и 17).
- Функции могут включать необязательные комментарии, описывающие аргументы и результаты: так, в результате объявления `def foo(x: "spam", y: list(range(3))) -> 42*2:` к объекту-функции на этапе времени выполнения будет присоединен атрибут-словарь `foo.func_annotations: {'x': "spam", 'y': [0, 1, 2], "return": 84}` (глава 15).
- Новая инструкция `nonlocal x, y` позволит выполнять присваивание переменным в области видимости функции (глава 16).
- Функция `apply(func, args, kws)` будет удалена. Вместо нее следует использовать синтаксическую конструкцию `func(*args, **kws)` (главы 16 и 17).

- Встроенная функция `reduce()` будет удалена. Вместо нее организуйте циклы, как показано в этой книге; `lambda`, `map()` и `filter()` будут сохранены в версии 3.0 (глава 17).
- Все импортирование по умолчанию будет выполняться по абсолютному пути, а собственный каталог пакета будет пропускаться: для выполнения импорта по относительному пути, как это делается сейчас, следует использовать синтаксическую конструкцию `from . import` (глава 21).
- Все классы будут классами нового стиля и будут поддерживать современные новые расширения (глава 26).
- Наследование `class Spam(object)`, необходимое для создания нынешних классов, будет необязательным для классов. В версии 3.0 и нынешняя «классика», и классы «нового стиля» будут относиться к тому, что сейчас называется классами нового стиля (глава 26).
- В инструкции `try` конструкция `except name` превратится в конструкцию `except name as value` (глава 27).
- В инструкции `raise` конструкция `raise E, V` должна будет записываться как `raise E(V)`, то есть экземпляр исключения должен создаваться явно (глава 27).
- Включен оператор контекста исключений `with/as`, описываемый в этой книге (глава 27).
- Все пользовательские и встроенные исключения описываются классами, а не строками (глава 28).
- Все исключения, определяемые пользователем, должны наследовать встроенный `BaseException` – корневой класс иерархии классов исключений (`Exception` – это его подкласс и его вполне можно использовать в качестве базового для организации своей иерархии). Встроенный класс `StandardException` будет ликвидирован (глава 28).
- Структура пакета стандартной библиотеки может быть существенно изменена (подробности – в примечаниях к выпуску Python 3.0).

Список нововведений может показаться устрашающим на первый взгляд, однако не следует забывать, что основы языка Python, описываемые в этой книге, останутся неизменными и в версии 3.0. В действительности большая часть из того, что перечислено выше, будет оказывать на программистов не слишком большое влияние, если это влияние вообще будет ощущаться.

Кроме того, следует отметить, что этот список по-прежнему остается достаточно гипотетическим и в конечном счете может оказаться неполным и неточным, поэтому за официальной информацией следует ознакомиться с примечаниями к выпуску Python 3.0. Если вы пишете программный код для линейки Python 2.0, обратите внимание на сценарий «2to3», который выполняет автоматическое преобразование программного кода, написанного для интерпретатора версии 2.x в программный

код для интерпретатора версии 3.0. Этот сценарий будет поставляться в составе Python 3.0.

Об этой серии

Книги издательства O'Reilly из серии «Изучаем...» предназначены для тех, кто хочет приобрести новые знания и предпочитает структурный подход к изучению. Каждая книга из этой серии использует принципы обучения, которые мы (не без вашей помощи) выработали для передачи знаний, необходимых вам, чтобы присоединиться к новому проекту, справиться с неожиданным заданием или быстро изучить новый язык программирования.

Чтобы получить максимум пользы от любой книги из серии «Изучаем...», мы рекомендуем прорабатывать главы последовательно. Вы увидите, что усвоение материала идет быстрее, если внимательно читать указания и рекомендации, включенные в них. Вы можете также использовать заключительные разделы для предварительного знакомства с ключевыми аспектами каждой главы и с тем, что вам предстоит изучать. Наконец, чтобы помочь вам проверить степень усвоения материала, каждая глава завершается разделом «Закрепление пройденного», который включает короткие контрольные вопросы. Кроме того, каждая часть включает практические упражнения.

Книги серии «Изучаем...» работают с вами как надежный и проверенный коллега или преподаватель; мы стремимся сделать ваше обучение как можно более приятным. Свои отзывы о нашей работе и предложения по улучшению можете направлять по адресу learning@oreilly.com.

Использование программного кода примеров

Данная книга призвана оказать вам помощь в решении ваших задач. Вы можете свободно использовать примеры программного кода из этой книги в своих приложениях и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Например, если вы разрабатываете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам *необходимо* получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, получение разрешения не требуется. Но при включении существенных объемов программного кода примеров из этой книги в вашу документацию вам *необходимо* будет получить разрешение издательства.

Мы приветствуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN. Например: «Learning Python, by Mark Lutz. Copyright 2008 O'Reilly Media, Inc., 978-0-596-51398-6».