

Предисловие

Во многом изучение языка программирования похоже на изучение иностранного (человеческого) языка. Вы можете пройти курс, понять предмет и даже хорошо сдать выпускной экзамен. Но, когда приходит время использовать язык на практике, вы можете оказаться в замешательстве, не зная, какой синтаксис использовать, или каким наиболее подходящим способ выразить свою мысль — не говоря уже о том, что вы не можете понять носителей языка.

Вот тут-то и приходит на помощь практика. Практика иностранного языка повышает уровень беглости и уверенности, позволяя вам вести более глубокие и интересные беседы. Практика Python позволит вам быстрее и проще решать проблемы и одновременно писать более читаемый и сопровождаемый код. Улучшение навыков происходит со временем, по мере того как вы используете язык в новых и разнообразных ситуациях. Зачастую легко не заметить свой прогресс. Но если вспомнить, как вы писали код всего несколько месяцев назад, разница будет очевидной.

Эта книга не предназначена для того, чтобы научить вас Python. Скорее, ее цель — дать вам практику, необходимую для свободного владения Python. После выполнения упражнений в этой книге — а не просто беглого просмотра вопросов и подглядывания в ответы — вы будете писать более читабельный, более правильный и удобный в сопровождении код на Python.

Упражнения на Python появились в результате бесед с моими студентами на корпоративных занятиях по обучению Python. После окончания курса они часто спрашивали, где можно найти

упражнения для дополнительной практики и улучшения своих навыков. Эта книга основана на практических занятиях, которые я провожу для своих студентов, а также на обсуждениях, возникающих во время и после занятий.

Упражнения призваны помочь вам усвоить некоторые из основных понятий языка Python: основные структуры данных, функции, генераторы, объектно-ориентированное программирование и итераторы. Эти темы могут показаться простыми, возможно, даже слишком простыми для книги упражнений. Но весь Python, от самого большого приложения до самого маленького скрипта, основан на этих фундаментальных вещах. Чтобы чувствовать себя уверенным при разработке на Python, очень важно знать эти основы. Я часто говорю, что игнорирование фундаментальных основ в пользу более сложных тем сродни тому, как если бы студент-химик игнорировал элементы в пользу «настоящих» химических веществ.

Я могу лично подтвердить важность практики не только как преподаватель Python, но и как студент. В течение нескольких лет я изучаю китайский язык, во многом из-за того, что каждые несколько месяцев я езжу в Китай для проведения курсов на Python. Каждый урок, который я беру, и каждое упражнение, которое я делаю, кажется, не сильно продвигает мое свободное владение языком. Но когда я возвращаюсь в Китай после нескольких месяцев отсутствия, я понимаю, что практика действительно помогла и что мне легче общаться с местными.

Мне еще далеко до свободного владения китайским языком, но я делаю успехи, и мне приятно оглядываться назад и видеть, как далеко я продвинулся. Я надеюсь и рассчитываю, что Упражнения на Python сделает то же самое для вас, улучшая ваше понимание и свободное владение языком с каждым днем.

Благодарности

Возможно, это клише, что написание книги является совместной работой, но тем не менее это чистая правда. Поэтому я хочу поблагодарить и выразить признательность людям, без которых этой книги не было бы.

Прежде всего я хочу поблагодарить тысячи студентов, которым я имел честь преподавать на моих корпоративных курсах по Python. Именно благодаря их вопросам, предложениям, соображениям и исправлениям решения и объяснения приобрели свой нынешний вид.

Спасибо также многим подписчикам моей еженедельной рассылки «Лучшие разработчики», которые часто находят время, чтобы прокомментировать и исправить темы, о которых я писал. Я многому научился у них и часто использую эти знания в своей преподавательской деятельности.

Далее Филип Гуо — доцент кафедры когнитивных наук Калифорнийского университета в Сан-Диего. Он также является автором и владельцем сайта Python Tutor, бесценного инструмента, который я часто использую в своих курсах и к которому я рекомендую своим студентам обращаться в случае, когда у них возникают сложности с написанием кода. В этой книге я использовал много скриншотов с Python Tutor, и почти каждое решение содержит ссылку на этот сайт, чтобы вы могли самостоятельно разобраться в коде.

Спасибо всем, кто работает над Python, начиная с основных разработчиков, тех, кто пишет и ведет блог о языке, и заканчивая теми, кто создает пакеты. Экосистема Python — это впечат-

ляющее технологическое достижение, но меня также поразило огромное количество по-настоящему отзывчивых, порядочных и дружелюбных людей, стоящих за этими достижениями.

Множество людей из Manning внесли свой вклад в эту книгу и сделали ее намного лучше, чем я бы сделал в одиночку. (И вот доказательство: прототип этой книги, опубликованный самостоятельно, был не так хорош, как тот, что вы сейчас читаете!) Я тесно сотрудничал с несколькими людьми, и все они благодаря мастерству и терпению помогли этой книге воплотиться в жизнь. Майкл Стивенс увидел перспективность книги, ориентированной на отработку упражнений, и предложил мне поработать с Manning. Фрэнсис Лефковиц не только умело отредактировала текст и указала, где его можно улучшить, разбить или проиллюстрировать; она также была со мной во время всего процесса написания книги. Гэри Хаббард и Игнасио Белтран Торрес дали мне бесчисленное количество технических советов и правок, находя ошибки и помогая улучшить неудачные объяснения. А Карл Кеснель произвел на меня неизгладимое впечатление своими подробными правками окончательного текста.

Всем рецензентам: Аннет Девинд, Билл Бейли, Чарльз Дэниелс, Кристоффер Финк, Дэвид Криф, Дэвид Моравек, Дэвид Р. Снайдер, Гэри Хаббард, Джефф Крейг, Глен Сиракавит, Жан-Франсуа Морен, Джефф Смит, Йенс Кристиан Б. Мадсен, Джим Амрхайн, Джо Юстесен, Киран Кут-Динь, Марк Элстон, Майур Патил, Мередит Годар, Стефан Трост, Стив Лав, Сушант Бхосале, Тамара А. Фульц, Тони Холдройд и Уоррен Майерс — ваши предложения помогли сделать эту книгу лучше.

Наконец, моя семья была терпелива на протяжении всей моей деловой и академической карьеры. Они поддерживали и помогали мне, пока я занимался развитием своих курсов, получал PhD и путешествовал по миру, преподавая Python. Они дважды проявили терпение касательно этой книги: когда я самостоятельно опубликовал ее на своем сайте, и когда она была обновлена, дополнена и улучшена (довольно значительно), чтобы стать тем, что вы сейчас читаете. Спасибо моей жене, Шире, и моим детям, Аtare, Шикме и Амоцу, за их понимание и признательность.

Об этой книге

Рpython Workout не предназначена для обучения вас Python, хотя я надеюсь и ожидаю, что вы узнаете много нового. Она призвана помочь вам улучшить понимание языка Python и его использование для решения проблем. Воспринимайте ее как сборник упражнений, силу и потенциал которого вы можете использовать. Чем больше усилий вы вложите в эту книгу, тем больше вы получите от нее.

Другими словами, это книга, которую нужно не просто прочитать или пролистать. Чтобы обучение прошло успешно, вам придется потратить время на поиск ответов на вопросы, неизбежно совершая ошибки. Есть большая разница между тем, чтобы прочитать о поиске решения, и тем, чтобы написать его самому. Я надеюсь, что вы потратите время на то, чтобы ответить на эти вопросы; я обещаю, что это с лихвой окупится в будущем.

За время работы с *Python Workout* вы решите множество задач, связанных с основными структурами данных, функциями, генераторами, модулями, объектами и итераторами. Вы поймете, как использовать их эффективно, и узнаете, как применять различные идиоматические способы. После выполнения этих упражнений вам будет легче разрабатывать и писать программы на Python для работы и для развлечения.

Обратите внимание, что не стоит искать помощи в документации Python или даже на таких сайтах, как Stack Overflow. Ни один разработчик не может запомнить все, что ему нужно в повседневной работе. Я надеюсь, что по мере изучения книги и дальнейшей работы с Python вы будете обращаться к подобной документации реже или только для ознакомления с более сложными темами.

Для кого эта книга

Эта книга предназначена для разработчиков, которые прошли курс Python или, возможно, прочитали вводную книгу по этому языку. Фактически большая часть этих упражнений предназначена для тех, кто проходит мой вводный курс по Python или недавно закончил его. Вы должны иметь представление об основных конструкциях, таких как `if` и `for`, а также об основных структурах данных, таких как строки, списки, кортежи и словари.

Но есть разница между мимолетным знакомством с этими темами и умением применять их для решения реальных проблем. Если вы умеете работать с Python, но каждый день по многу раз обращаетесь к Stack Overflow, то эта книга поможет вам стать более уверенным и независимым в написании кода на Python. Думаю, что если вы регулярно программируете на Python не менее шести месяцев, то эта книга будет вам полезна.

Из чего состоит эта книга: дорожная карта

Эта книга состоит из десяти глав, каждая из которых посвящена отдельным аспектам Python. Однако в упражнениях каждой главы будут использоваться методы из других глав. Например, почти в каждом упражнении вас попросят написать функцию или класс, хотя функции представлены в главе 6, а классы — в главе 9. Воспринимайте эти названия как общие рекомендации, а не строгие правила для того, что вы будете практиковать и изучать в каждой главе.

Главы

1. **Числовые типы:** Целые числа и числа с плавающей запятой — преобразования значений между числами и строками.
2. **Строки:** Работая со строками, рассматривайте их не только как текст, но и как последовательности, которые можно итерировать.

3. **Списки и кортежи:** Создание, изменение (в случае списков) и извлечение из списков и кортежей.

4. **Словари и множества:** Различные способы использования словарей и некоторых их полезных методов. Кроме того, в некоторых случаях используются множества, связанные со словарями.

5. **Файлы:** Чтение и запись в файл.

6. **Функции:** Написание функций, включая вложенные функции. Изучение области видимости переменных в Python.

7. **Функциональное программирование с генераторами:** Решение задач при помощи списков, множеств и генераторов словарей.

8. **Модули и пакеты:** Написание и использование модулей в Python-программе.

9. **Объекты:** Создание классов, написание методов, использование атрибутов и объяснение наследования.

10. **Итераторы и генераторы:** Добавление протокола итератора в классы, написание функций-генераторов и представлений генераторов.

Об этой книге

Упражнения составляют основную часть каждой главы. Каждое упражнение состоит из пяти компонентов:

1. **Упражнение:** Постановка проблемы, которую вам предстоит решить.

2. **Обсуждение:** Подробное обсуждение проблемы и способов ее решения.

3. **Решение:** Код решения, а также ссылка на код на сайте Python Tutor [qr1], чтобы вы могли его запустить. Код доступен на GitHub [qr2].

4. **Скринкаст решения:** Короткое видео, представляющее собой запись экрана с объяснением решения. В видео будет показан



не только ответ, но и процесс поиска решения. При чтении книги на liveBook, платформе Manning, видео появляются сразу после каждого решения. В печатной и электронной книге есть ссылка на навигационную страницу [qr3], нужное упражнение можно найти по номеру и названию.

5. После выполнения упражнения:

Три дополнительных, связанных между собой задания. На эти задания нет ни ответов, ни обсуждений в книге, но вы можете скачать соответствующий код. (Подробности см. в следующем разделе.) Вы можете обсудить эти дополнительные задания и сравнить решения с другими читателями Python Workout на онлайн-форуме, посвященном книге, платформы Manning — liveBook.



В дополнение к упражнениям приводятся многочисленные сноски со справочной информацией, в которых объясняется тема, часто ставящая разработчиков Python в тупик. Например, в сносках рассказывается про f-строки, определение области видимости переменной и о том, что происходит при создании нового объекта. В книге также содержится множество подсказок, советов и примечаний — все они призваны помочь вам улучшить навыки программирования на Python и предостеречь от повторения ошибок, которые я неоднократно совершал на протяжении многих лет.

О коде

В этой книге содержится большое количество кода на языке Python. В отличие от большинства книг, вы должны скорее написать код, чем просто прочитать его. Возможно, некоторые читатели (возможно, вы!) придумают решения лучше, чем у меня, более правильные или более элегантные. Если это так, то не стесняйтесь написать мне об этом.

Решения всех упражнений, включая задачи из «После выполнения упражнения», доступны в двух местах: на сайте Python

Tutor, который предоставляет среду для выполнения кода, или на GitHub по адресу, где вы можете скачать код. Этот репозиторий не только содержит все решения, но и включает тесты `pytest` для каждого из них. (Не знакомы с `pytest`? Я настоятельно рекомендую вам прочитать о нем здесь [qr4] и использовать его для проверки вашего кода.)



Между кодом в репозитории GitHub и тем, что опубликовано в книге, есть небольшие различия. В частности, решения в книге не включают строки документации для функций, классов и модулей, в отличие от репозитория.

Эта книга содержит множество примеров исходного кода как в пронумерованных листингах, так и в строках обычного текста. В обоих случаях исходный код оформляется моноширинным шрифтом, в котором все знаки имеют одинаковую ширину, чтобы выделить его в тексте. Иногда код также выделяется **жирным шрифтом**, чтобы подчеркнуть изменения в коде, например, добавление новой функции к существующей строке кода.

Во многих случаях исходный код был переформатирован; мы добавили переносы строк и изменили отступы, чтобы уместить их на свободных страницах книги. В редких случаях этого было недостаточно, и мы включили в списки маркеры перевода строки (¶). Кроме того, комментарии в исходном коде часто удалялись из листингов, если код описывался в тексте. В большинстве листингов содержатся аннотации к коду, подчеркивающие важные концепции.

Как я уже говорил ранее, приобретая эту книгу, вы также получаете доступ к скринкастам, в которых я показываю решения упражнений. Я надеюсь, что благодаря решениям задач (в печатном виде), объяснениям, ссылкам на Python Tutor, коду из репозитория, тестам `pytest` и скринкастам, вы сможете в полной мере понять каждое решение и применить его в своем собственном коде.

Требования к программному/ аппаратному обеспечению

Прежде всего, вам нужно установить python. Загрузить и установить его проще всего с сайта [qr5]. Я рекомендую установить последнюю доступную версию. Существуют также альтернативные способы установки Python, включая Windows Store или Homebrew для Mac.



Для работы с этой книгой подойдет любая версия Python 3.6 и выше. В нескольких местах текст описывает возможности, которые являются новыми в Python 3.7 и 3.8, но все решения используют методы, которые работают с 3.6. Все программы работают в разных операционных системах, поэтому независимо от того, какую платформу вы используете, упражнения в этой книге будут работать.

Технически вам не нужно устанавливать редактор или IDE (интегрированную среду разработки) для Python, но они обязательно пригодятся. Двумя наиболее популярными IDE являются PyCharm (от JetBrains) и VSCode (от Microsoft). Более консервативные разработчики Python используют vim или Emacs (мой личный фаворит). Но в конце концов, вы можете (и должны) использовать тот редактор, который вам больше подходит. Языку Python неважно, какую IDE вы используете.

Форум для обсуждений liveBook

Покупая книгу *Python Workout*, вы получаете бесплатный доступ к частному веб-форуму Manning Publications, где вы можете оставить комментарии о книге, задать технические вопросы и получить помощь от автора и других пользователей. Чтобы получить доступ к форуму, перейдите по этому адресу [qr6]. Вы также можете



узнать больше о форумах Manning и правилах поведения на сайте [qr7].

Manning предоставляет площадку, где наши читатели могут обсудить интересующие их вопросы, а также пообщаться с автором. Это не накладывает обязательства на автора участвовать в обсуждении, его вклад в форум остается добровольным (и неоплачиваемым). Мы предлагаем вам попробовать задать автору несколько сложных вопросов, чтобы его интерес не пропал! Форум и архивы предыдущих обсуждений будут доступны на сайте издательства до тех пор, пока книга находится в печати.

