

ПРЕДИСЛОВИЕ

Третье издание книги *Python: создание приложений!*

Мы рады, что вы обратились к нам за помощью в максимально быстром и глубоком изучении языка Python. Цель серии *Core Python* не сводится к простому преподаванию языка Python; мы хотим, чтобы вы получили такой уровень знаний, чтобы разрабатывать программное обеспечение в любой прикладной области.

В других книгах серии *Core Python* — *Core Python Programming* и *Core Python Language Fundamentals* — мы не только описываем синтаксис этого языка, но и стремимся всесторонне изложить его структуру. Мы полагаем, что вооружившись этими знаниями, вы напишете более эффективные приложения, независимо от уровня вашей подготовки.

После прочтения любой другой вводной книги о языке Python может показаться, что вы изучили его достаточно хорошо. Выполняя многочисленные упражнения, вы, вероятно, даже вполне уверены в своих навыках программирования на языке Python. Однако у вас могут возникнуть вопросы: “И что теперь? Какие виды приложений я могу создать с помощью языка Python?” Если вы изучали язык Python для создания узкоспециализированного проекта, то можете спросить: “Что еще я могу разработать с помощью языка Python?»

О книге

В книге *Python: создание приложений* вы узнаете обо всем, что следует знать о языке Python, и получите новые навыки, позволяющие создавать разнообразные приложения.

Эти главы повышенной сложности предназначены для “быстрого погружения” в разнообразные темы. Если вы углубитесь в определенные области разработки приложений, охваченные в какой-либо из этих глав, то, вероятно, обнаружите, что они содержат более чем достаточно информации, чтобы направить вас в правильном

направлении. *Не ожидайте* всестороннего изложения, потому что это умалило бы универсальность данной книги.

Как и во всех других книгах серии *Core Python*, в этой книге приведено много примеров, которые можно проверить на вашем компьютере. Для того чтобы закрепить усвоенные понятия, в конце каждой главы приводятся как простые, так и сложные упражнения. Они предназначены для проверки ваших знаний и навыков программирования на языке Python. Практический опыт ничем невозможно заменить. Мы полагаем, что вы должны не только получить навыки программирования на языке Python, но и усвоить их за максимально короткий период времени.

Поскольку лучший способ получить навыки — это практика, упражнения представляют собой одно из самых больших преимуществ этой книги. С их помощью вы можете проверить свои знания, полученные из глав, а также получить опыт программирования. Для закрепления навыков нет ничего эффективнее, чем разработка приложений. Вам придется решать легкие, средние и трудные проблемы. По просьбе читателей мы включили в книгу задачи, подразумевающие необходимость писать большие, а не игрушечные и практически бесполезные программы. Ответы на некоторые упражнения приведены в приложении А, а справочные таблицы — в приложении Б.

Я хотел бы поблагодарить всех читателей за советы и предложения. Именно благодаря вам я стал писать книги. Я прошу вас писать мне письма и помочь подготовить *четвертое* издание еще лучше, чем все предыдущие!

Для кого предназначена эта книга

Для всех, кто знает о существовании языка Python и хочет знать больше, развивая свои навыки программирования приложений. Язык Python применяется во многих областях, включая промышленность, информационные технологии, науку, бизнес, индустрию развлечений и др. Это значит, что список пользователей языка Python (и читателей этой книги) включает следующие профессии (но не ограничивается ими):

- разработчики программного обеспечения;
- разработчики систем автоматизированного проектирования;
- разработчики систем контроля качества и средств автоматизации;
- разработчики информационных систем и сетевые администраторы;
- ученые и математики;
- проектировщики и менеджеры, управляющие проектами;
- разработчики мультимедийных и аудиовизуальных систем;
- менеджеры, управляющие логистическими цепочками, и выпускающие менеджеры;
- веб-мастера и штат управления контентом;
- инженеры технической поддержки;
- разработчики и администраторы баз данных;
- инженеры, принимающие участие в научно-исследовательских проектах;
- инженеры, которые занимаются интеграцией и обслуживанием программного обеспечения;

- университетские преподаватели;
- разработчики веб-служб;
- разработчики финансового программного обеспечения;
- и многие другие!

Список знаменитых компаний, использующих язык Python, включает: Google, Yahoo!, NASA, Lucasfilm/Industrial Light and Magic, Red Hat, Zope, Disney, Pixar и Dreamworks.

Автор и язык Python

Я открыл для себя язык Python примерно десять лет назад, работая в компании Four11. В то время основным продуктом этой компании была служба каталогов Four11.com White Page. Язык Python использовался для разработки следующего продукта: веб-службы электронной почты Rocketmail, которая в итоге эволюционировала в службу Yahoo! Mail.

Изучать язык Python и разрабатывать службу Yahoo! Mail было интересно. Я помог перепроектировать адресную книгу и механизм проверки правописания. В то время язык Python стал частью многих других сайтов Yahoo!, включая службы People Search, Yellow Pages, а также Maps and Driving Directions. Фактически я был ведущим разработчиком службы People Search. Хотя в то время я плохо знал язык Python, выучить его было очень легко — он много проще других языков, которые я осваивал прежде. Из-за дефицита учебников, существовавшего в то время, в качестве основных источников знаний мне пришлось использовать справочники *Library Reference* и *Quick Reference Guide*; это и стало главной мотивацией для написания книги, которую вы читаете сейчас. Со времени моей работы в компании Yahoo! я мог использовать язык Python для решения любых интересных задач. В каждом из этих случаев я смог использовать мощь языка Python для своевременного решения поставленной задачи. Я также разработал несколько курсов по языку Python и использовал эту книгу для преподавания. Книги серии *Core Python* превосходно помогают не только *учиться*, но и *учить*. Как инженер я знаю, как сложно изучить, освоить и применить новую технологию. Как профессиональный преподаватель я также знаю то, что необходимо для *эффективного обучения учеников*. Эти книги дают опыт, необходимый для выявления реальных аналогий и подсказок, которые невозможно получить у “простого учителя” или «обычного книжного автора”.

Особенности стиля изложения: технический, но простой

Мой опыт показывает, что для успешного и быстрого освоения языка Python нужна не книга для новичков или справочник по основам информатики, а книга, ориентированная на технические подробности. Для того чтобы ускорить процесс обучения, мы будем вводить теоретические понятия, иллюстрируя их соответствующими примерами. В конце каждой главы вы найдете многочисленные упражнения, закрепляющие некоторые изложенные понятия и идеи.

Мы не стремимся конкурировать со стилем Брюса Эккеля (Bruce Eckel) (см. рецензии на первое издание на веб-сайте <http://corepython.com>). Это не сухой университетский учебник. Наша цель состоит в том, чтобы разговаривать с вами, как будто вы сидите в аудитории. Как вечный студент я постоянно ставлю себя на место своих учеников и говорю им то, что им следует слышать, чтобы изучить понятия настолько быстро и полно, насколько это возможно. Чтение этой книги вам покажется быстрым и легким, но при этом вы не упустите из виду технические детали.

Как инженер я знаю то, что должен сказать вам, чтобы объяснить понятие из языка Python. Как учитель могу взять технические детали и перевести их на язык, который легко понять и моментально усвоить. Вы можете извлечь из этого то, что принесет вам наибольшую пользу, и при этом вы еще больше полюбите программировать на языке Python.

Как вы могли заметить, несмотря на то, что я — единственный автор книги, я использую третье множественное число, т.е. говорю “мы” и “наш”, потому что мы вместе двигаемся к общей цели — освоению языка Python.

О третьем издании

В момент выхода первого издания книги язык Python вступал в свою вторую эру вместе с выпуском версии 2.0. С тех пор язык подвергся существенным улучшениям, которые способствовали полному дальнейшему успеху, принятию и росту его популярности. В нем были устранены недостатки и добавлены новые функциональные возможности, отражающие новый уровень его мощи и богатый опыт многочисленных разработчиков.

Второе издание книги вышло в 2006 году, в разгар господства языка Python, одновременно с появлением его самой популярной версии 2.5. Второе издание вызвало восторженные отзывы и превзошло по продажам первое. Язык Python получил многочисленные награды, в частности:

- Tiobe (www.tiobe.com)
 - Язык года (2007, 2010)
- LinuxJournal (linuxjournal.com)
 - Лучший язык программирования (2009–2011)
 - Лучший язык описания сценариев (2006–2008, 2010, 2011)
- Премии членов сообщества LinuxQuestions.org
 - Язык года (2007–2010)

Эти награды еще больше способствовали популярности языка Python.

В настоящее время появилось третье поколение — версия Python 3. Книга *Core Python Programming* тоже перешла на третий этап, и я благодарен издательству Prentice Hall за предложение написать ее третье издание.

Поскольку версия 3.x не имеет обратной совместимости с версиями Python 1 и 2, для ее полной адаптации в промышленности потребуется определенное время. Мы рады помочь вам совершить этот переход. Программы в этом издании будут представлены как в версии Python 2, так и в версии Python 3 (только по возможности, поскольку пока не все можно перенести из одной версии в другую). Мы также обсудим различные инструменты и методы, необходимые для перехода.

Изменения, внесенные в версии 3.x, продолжают тенденцию к улучшению языка и позволяют сделать большой шаг к удалению некоторых его последних недостатков, а также большой скачок в развитии языка. Структура книги также значительно изменилась. Из-за своего размера и объема книга Core Python Programming требовала пересмотра для выпуска третьего издания.

По этой причине издательство Prentice Hall и я решили, что лучше всего оставить логическую структуру частей I и II предыдущих изданий, в которых представлено ядро языка и вопросы, связанные с разработкой сложных приложений соответственно, и разделить книгу на два тома.

Вы держите в руках вторую часть третьего издания книги Core Python Programming. Хорошая новость заключается в том, что для понимания второй части первая часть не требуется. Мы лишь хотели бы, чтобы читатели имели умеренный опыт по программированию на языке Python. Если вы лишь недавно изучили язык Python и стали программировать на нем или уже имеете опыт и хотите перейти на новый уровень, то вы нашли то, что нужно!

Читатели предыдущих изданий книги Core Python Programming уже знают, что мы подробно излагаем основные вопросы, не ограничиваясь синтаксисом (разве для изучения синтаксиса нужна книга?). Зная, как устроен язык Python, включая отношение между объектами данных и механизмом управлением памятью, можно стать более эффективным программистом. Эти вопросы остались в первой части и стали основой книги Core Python Language Fundamentals.

Как и прежде, я продолжаю вести веб-сайт и блог, а также писать статьи, чтобы информация оставалась как можно более свежей независимо от версии языка Python. В новое издание добавлены следующие темы.

- Примеры реализации электронной почты с помощью веб (глава 3).
- Использования библиотеки Tile/Ttk (глава 5).
- Использование базы данных MongoDB (глава 6).
- Более подробные примеры использования программ Outlook и PowerPoint (глава 7).
- Стандарт WSGI (глава 10).
- Использование социальной сети Twitter (глава 13).
- Использование службы Google+ (глава 15).

Кроме того, мы рады предложить три совершенно новые главы: главу 11, “Веб-платформы: Django”; главу 12, “Облачные вычисления: Google App Engine”, и главу 14, “Обработка текста”. В них описаны новые или существующие области применения языка Python. Все существующие главы были обновлены с учетом последних версий языка Python, включая новый материал. В начале каждой главы приводится список тем, которые в ней рассматриваются.

Путеводитель по главам

Книга разделена на три части. Первая часть, занимающая почти две трети всего объема, посвящена основным инструментам, входящим в любой пакет для разработки приложений (с упором на язык Python, разумеется).

Во второй части излагаются разнообразные темы, связанные с веб-программированием. Завершается книга экспериментальными главами, которые касаются совершенно новых вопросов и будут уточняться в следующих изданиях. Все три части охватывают одну тему — создание приложений на языке Python. Мы рады помочь читателям разобраться во многих ключевых аспектах программирования на языке Python.

Ниже приводится подробное описание глав.

Часть I. Общие прикладные темы

Глава 1. Регулярные выражения

Регулярные выражения — мощный инструмент, позволяющий выявлять шаблоны, извлекать их, а также выполнять операции поиска и замены.

Глава 2. Сетевое программирование

В настоящее время многие приложения связаны с работой в сетях. В этой главе мы научим вас создавать клиентов и серверы с помощью протоколов TCP/IP и UDP/IP, а также расскажем, что такое SocketServer и Twisted.

Глава 3. Программирование интернет-клиентов

Большинство интернет-протоколов, используемых в настоящее время, разработаны с помощью сокетов. В главе 3 мы исследуем высокоуровневые библиотеки, используемые для создания клиентов этих интернет-протоколов. В частности, мы изучим протокол передачи файлов (FTP), протокол передачи новостей Usenet (NNTP) и множество протоколов электронной почты (SMTP, POP3, IMAP4).

Глава 4. Многопоточное программирование

Многопоточное программирование — один из способов улучшить производительность приложения с помощью параллельной работы. Эта глава посвящена вопросам реализации потоков на языке Python. В ней объясняются основные понятия и показывается, как правильно создать многопоточное приложение на языке Python и какие сценария использования являются лучшими.

Глава 5. Программирование графического пользовательского интерфейса

Набор графических инструментов Tkinter (в версии Python 3 переименованный в `tkinter`), основанный на библиотеке Tk, по умолчанию является основной библиотекой для разработки графического пользовательского инструмента на языке Python. С его помощью мы покажем, как создать простые приложения с графическим пользовательским интерфейсом. Один из лучших способов обучения — копирование. Используя эти приложения, вы быстро освоите эту тему. В главе кратко рассмотрены и другие графические библиотеки, такие как Tix, Pmw, wxPython, PyGTK и Tk/Tile.

Глава 6. Программирование баз данных

Язык Python позволяет упростить программирование баз данных. Сначала мы рассмотрим основные понятия, а затем изучим интерфейс программирования приложений, управляющих базами данных (DB-API). После этого будет показано, как установить соединение с реальной базой данных и выполнить запросы и другие операции с помощью языка Python. Если Вы предпочитаете автоматический подход, основанный на языке Structured Query Language (SQL), и просто хотите работать с объектами, не интересуясь устройством базы данных, то можете использовать объектно-реляционные преобразователи. В заключение мы вводим читателей в мир нереляционных баз данных, экспериментируя с базой данных MongoDB, играющей роль примера технологии NoSQL.

Глава 7. Программирование приложений для работы с Microsoft Office

Нравится Вам это или нет, но мы живем в мире, в котором нам приходится вступать в контакт с персональными компьютерами, работающими под управлением операционной системы Microsoft Windows. Независимо от того, насколько часто мы это делаем, язык Python может облегчить нашу жизнь. В этой главе исследуется программирование клиентов COM с помощью языка Python для управления и взаимодействия с приложениями Office, такими как Word, Excel, PowerPoint и Outlook. В предыдущих изданиях эта глава была экспериментальной, а теперь она расширена и стала самостоятельной.

Глава 8. Создание расширений для языка Python

Ранее мы уже подчеркивали, насколько мощными являются повторное использование кода и расширения языка. В языке Python этими расширениями являются модули и пакеты. Однако помимо этого существует возможность разработки низкоуровневых программ на языках C/C++, C# и Java. Эти расширения легко интегрируются в язык Python. Создание расширений на низкоуровневых языках позволяет повысить производительность приложений и усилить безопасность (поскольку исходный код раскрывать необязательно). В этой главе шаг за шагом рассматривается процесс расширения языка Python с помощью языка C.

Часть II. Разработка веб-приложений

Глава 9. Веб-клиенты и веб-серверы

Продолжая обсуждение клиент-серверной архитектуры, начатое в главе 2, мы применяем эту концепцию к веб. В этой главе мы изучим разнообразные инструменты веб-клиентов, методы анализа веб-контента и, наконец, покажем, как написать свой веб-сервер на языке Python.

Глава 10. Веб-программирование: интерфейсы CGI и WSGI

Главная работа веб-сервера состоит в том, чтобы получать запросы клиента и возвращать результаты. Но как серверы получают эти данные? Несмотря на то что они действительно хороши при возвращении результатов, у них обычно нет возможностей или логических механизмов, необходимых для этого; основная работа выполняется в другом месте. Интерфейс CGI позволяет серверам запускать другие программы, выполняющие эту обработку, однако он не допускает масштабирования и поэтому не используется на практике; тем не менее его концепции все еще применяются, независимо от того, какую платформу вы используете. Таким образом, большая часть главы посвящена изучению интерфейса CGI. Вы также узнаете, как интерфейс WSGI помогает разработчикам приложений, предоставляя им общий программный интерфейс. Кроме того, вы увидите, как интерфейс WSGI помогает разработчикам платформ, которые должны соединять веб-серверы на одной стороне с кодом приложения на другом, работать так, чтобы разработчики приложений могли не беспокоиться о платформе выполнения.

Глава 11. Веб-платформы: Django

Язык Python обеспечивает разработку веб-приложений с помощью платформы Django, ставшей одной из самых популярных. В этой главе будет описан этот каркас и показано, как написать простые веб-приложения. Эти знания позволят вам при необходимости приступить к изучению других веб-платформ.

Глава 12. Облачные вычисления: Google App Engine

Облачные вычисления покоряют индустрию. Миру широко известны инфраструктурные службы AWS компании Amazon и онлайн-приложения, такие как Gmail и Yahoo! Mail. Однако существуют мощные альтернативные платформы, позволяющие

не нагружать пользователей лишними знаниями об инфраструктуре и предоставляющие им больше гибкости благодаря управлению их приложениями и кодом. В этой главе содержится исчерпывающее введение в первую службу на языке Python — Google App Engine. Обладая этими знаниями, Вы сможете изучить другие аналогичные службы.

Глава 13. Веб-службы

В этой главе изучаются высокоуровневые веб-службы (основанные на протоколе HTTP). Мы опишем как старую службу (Yahoo! Finance), так и новую (Twitter). Вы научитесь работать с обеими службами, используя язык Python и знания, полученные в предыдущих главах.

Часть III. Дополнительная и экспериментальная

Глава 14. Обработка текста

Первая дополнительная глава посвящена вопросам обработки с помощью языка Python. Сначала мы исследуем формат CSV, затем JSON и, наконец, XML. В конце главы мы воспользуемся клиент-серверной архитектурой в сочетании с языком XML для создания службы вызова удаленных процедур (RPC) с помощью протокола XML-RPC.

Глава 15. Разное

Эта глава содержит дополнительный материал, который будет разработан в следующем издании. Она посвящена интерпретатору Java/Jython и службе Google+.

Соглашения

Все результаты работы программ и исходные коды выделены с помощью моноширинного шрифта. Ключевые слова языка Python выделены **полужирным моноширинным** шрифтом. Строки вывода начинаются с трех символов “больше” (>>>), обозначающих приглашение интерпретатора Python. Звездочка (*) перед названием главы, раздела или упражнения означает сложный и/или факультативный материал.



Примечание



Модуль



Подсказка

2.5

Номер версии, в которой появились новые возможности языка Python

Сайты, посвященные книге

Мы приветствуем любые замечания — хорошие, плохие и ужасные. Если у вас есть комментарии, предложения, комплименты, жалобы, указания на ошибки, вопросы и что-нибудь еще, пишите по адресу corepython@yahoo.com.

На веб-сайте <http://corepython.com> вы найдете список замеченных ошибок, исходные коды, обновления, анонсы, упражнения, файлы для загрузки и другую информацию о книге. Вы можете принять участие в обсуждении книг серии *Core Python* на странице службы Google+ по адресу <http://plus.ly/corepython>.

Благодарности рецензентам и соавторам третьего издания

Глория Вилладсен (Gloria Willadsen) (главный рецензент)
 Мартин Омандер (Martin Omander) (рецензент и соавтор главы 11, “Веб-платформы: Django,” создатель приложения TweetApprover и соавтор раздела 15.2 “Служба Google+” в главе 15, “Разное”).
 Дарлин Вонг (Darlene Wong)
 Брайм Вердьер (Bryce Verdier)
 Эрик Вальстад (Eric Walstad)
 Поль Биссекс (Paul Bissex) (соавтор книги Python Web Development with Django)
 Йохан “proppy” Ефросин (Johan “proppy” Euphrosine)¹
 Энтони Валлон (Anthony Vallone)

Вдохновители

Моя жена Фей (Faye), которая продолжает поражать меня способностью управлять домашним хозяйством, заботиться о детях и их графике, кормить нас всех, вести финансы и в состоянии сделать это, в то время как я витаю в облаках или пишу книги.

Редакционная коллегия

Марк Тауб (Mark Taub) (главный редактор)
 Дебра Уильямс Кэли (Debra Williams Cauley) (рецензент издательства)
 Джон Фуллер (John Fuller) (ответственный редактор)
 Элизабет Райан (Elizabeth Ryan) (редактор проекта)
 Боб Рассел, Octal Publishing, Inc. (литературный редактор)
 Дайан Рассел, Octal Publishing, Inc. (технический редактор)

Благодарности ко второму изданию

Рецензенты и соавторы

Шаннон -jj Беренс (Shannon -jj Behrens) (главный рецензент)
 Майкл Сантом (Michael Santos) (главный рецензент)
 Рик Кван (Rick Kwan)
 Линделл Алдерманн (Lindell Aldermann) (один из авторов раздела о Юникоде в главе 6)
 Вей-Йип Тун (Wai-Yip Tung) (один из авторов примера о Юникоде в главе 20)
 Эрик Фостер-Джонсон (Eric Foster-Johnson) (соавтор по книге Beginning Python)
 Алекс Мартелли (Alex Martelli) (редактор книги Python Cookbook и автор книги Python in a Nutshell)
 Ларри Розенштейн (Larry Rosenstein)
 Джим Орош (Jim Orosz)
 Кришна Сринивасан (Krishna Srinivasan)
 Чак Кунг (Chuck Kung)

¹ Proppy — псевдоним Йохана Ефросина в социальных сетях.

Вдохновители

Мои прекрасные дети и хомяк.

Благодарности к первому изданию

Рецензенты и соавторы

Гвидо ван Россум (Guido van Rossum) (создатель языка Python)
 Доусон Тонг (Dowson Tong)
 Джеймс С. Алстром (James C. Ahlstrom) (соавтор по книге Internet Programming with Python)
 С. Канделария де Рам (S. Candelaria de Ram)
 Кэй С. Хорстманн (Cay S. Horstmann) (соавтор по книге Core Java и Core JavaServer Faces)
 Майкл Сантом (Michael Santos)
 Грэг Уард (Greg Ward) (создатель пакета distutils и его документации)
 Винсент С. Рубино (Vincent C. Rubino)
 Мартийн Фаассен (Martijn Faassen)
 Эмиль ван Себил (Emile van Seville)
 Реймонд Цай (Raymond Tsai)
 Алберт Л. Андерс (Albert L. Anders) (один из авторов главы о проекте MT Programming)
 Фредрик Лунд (Fredrik Lundh) (автор книги Python Standard Library)
 Камерон Лейрд (Cameron Laird)
 Фред Л. Дрейк, мл. (Fred L. Drake, Jr.) (соавтор по книге Python & XML и редактор официальной документации по языку Python)
 Джереми Хилтон (Jeremy Hylton)
 Стив Йошимото (Steve Yoshimoto)
 Ааз Марух (Aahz Maruch) (автор по книге Python for Dummies)
 Джеффри Е.Ф. Фридл (Jeffrey E. F. Friedl) (автор книги Mastering Regular Expressions)
 Питер Клерхут (Pieter Claerhout)
 Катриона (Кейт) Джонстон (Catriona (Kate) Johnston)
 Дэвин Эшер (David Ascher) (соавтор по книге Learning Python и редактор книги Python Cookbook)
 Рег Чарни (Reg Charney)
 Кристиан Тисмер (Christian Tismer) (создатель версии Stackless Python)
 Джейсон Стилвелл (Jason Stillwell)
 и мои студенты из филиала Калифорнийского университета в Санта-Крузе (UC Santa Cruz Extension)

Вдохновители

Я хотел бы выразить глубокую благодарность Джеймсу П. Прайору (James P. Prior), моему университетскому преподавателю по программированию.

Огромное спасибо Луизе Мозер (Louise Moser) и Майклу Меллиар-Смиту (P. Michael Melliar-Smith) (моим научным руководителям во время работы над диссертацией в Калифорнийском университете в Санта-Барбаре).

Благодарю Алана Парсонса (Alan Parsons), Эрика Вулфсона (Eric Woolfson), Эндрю Пауэлла (Andrew Powell), Яна Бейрсона (Ian Bairnson), Стюарта Эллиотта (Stuart Elliott), Дэвида Пэйтона (David Paton) и других участников проекта, а также членов форумов Projectologist и Roadkiller (за музыку, поддержку и приятное времяпрепровождение).

Я хотел бы поблагодарить свою семью, друзей и Господа Бога, уберегшего и наставившего меня на верный путь. Я благодарен всем, кто верил в меня на протяжении последних двадцати лет (вы знаете, кто я такой!), — я ничего не смог бы сделать без вас.

И в заключение хотел бы поблагодарить вас, мои читатели, и сообщество любителей языка Python. Я рад научить вас программированию на языке Python и надеюсь, что вы получите удовольствие на протяжении нашего, уже третьего, путешествия.

Уэсли Дж. Чан (Wesley J. Chun)
Силиконовая Долина, Калифорния (Silicon Valley, CA)
(Это не столько место, сколько состояние душевного здоровья.)
Октябрь 2001 г.; обновлено в июле 2006 г.,
марте 2009 г. и марте 2012 г.