

Введение

Язык программирования Python обладает уникальными возможностями, осознать которые поначалу не так-то просто. Предыдущий опыт многих программистов, знакомых с другими языками, часто мешает им в полной мере оценить выразительность средств Python и эффективно использовать предлагаемые возможности. Нередко встречаются программисты, которые впадают в другую крайность, проявляя чрезмерное усердие в попытках выжать из Python все, что только возможно. Однако при отсутствии надлежащего опыта такой подход может порождать серьезные проблемы, с которыми впоследствии приходится бороться.

В книге тщательно анализируется стиль написания программ, который принято характеризовать фразой *в духе Python* (Pythonic way) и который позволяет максимально эффективно задействовать все возможности этого языка. Автор предполагает, что основы Python читателю уже известны. Новички ознакомятся здесь с наилучшими методиками применения Python. Опытные программисты научатся уверенно применять новые для них и поэтому непривычные на первых порах инструменты этого языка.

Моя задача заключается в том, чтобы подготовить вас к эффективному использованию языка Python.

О чем эта книга

Каждая глава представляет собой сборник рекомендаций, объединенных в рамках одной темы. Можете читать их в произвольном порядке, выбирая то, что в данный момент вас больше всего интересует. Рекомендации включают советы относительно того, как решить ту или иную задачу, избежать типичных ошибок или обеспечить разумный компромисс в спорных моментах, а также позволяют осознанно подойти к выбору вариантов решений, являющихся наилучшими в тех или иных ситуациях.

Приведенные в книге рекомендации адресованы программистам, работающим с версиями Python 3 и Python 2. Большинство рекомендаций подойдет также программистам, использующим альтернативные варианты реализации Python, такие как Jython, IronPython, PyPy и др.

Глава 1. Мыслим категориями языка Python

В Python-сообществе бытует выражение “в духе Python”. Так говорят о программах, выдержанных в стиле, максимально эффективно задействующем особенности этого языка. Идиомы Python выработаны на основе живого практического опыта программистов, использующих этот и другие языки. В этой главе описаны наилучшие подходы к решению задач, с которыми часто сталкиваются программисты, работающие с языком Python.

Глава 2. Функции

Функции в языке Python обладают целым рядом дополнительных особенностей, облегчающих жизнь программистам. Некоторые из них имеют аналоги в других языках, но многие свойственны только Python. Эта глава посвящена использованию функций для раскрытия намерений программиста, содействия повторному использованию кода и уменьшения вероятности того, что в него вкрадутся ошибки.

Глава 3. Классы и наследование

Python — объектно-ориентированный язык. Чтобы что-то сделать в Python, часто требуется написать классы и определить способы их взаимодействия через интерфейсы и иерархию. В этой главе рассказывается о том, как использовать классы и наследование для создания объектов с определенным поведением.

Глава 4. Метаклассы и атрибуты

Метаклассы и динамические атрибуты — мощные средства Python. В то же время они позволяют реализовывать чрезвычайно причудливые и неожиданные варианты поведения. В этой главе рассмотрены наиболее распространенные идиомы для использования этих механизмов, гарантирующие соблюдение принципа *наименьшей неожиданности*.

Глава 5. Одновременность и параллелизм

Язык Python упрощает написание программ, обеспечивающих одновременное выполнение нескольких вычислительных задач. Кроме того, Python может быть использован для организации параллельных вычислений посредством системных вызовов, подпроцессов и C-расширений.

В этой главе обсуждается наилучшая практика применения Python в подобных ситуациях.

Глава 6. Встроенные модули

Пакет Python включает много важных модулей, которые понадобятся вам при написании программ. Стандартные модули настолько тесно переплетены с идиоматикой Python, что могли бы выступать в качестве составной части спецификации языка. В этой главе рассмотрены наиболее важные из встроенных модулей.

Глава 7. Совместная работа

Совместная работа над программами на языке Python требует особенно внимательного отношения к написанию кода. Но даже если вы работаете один, вам все равно надо знать, как использовать модули, написанные другими разработчиками. Эта глава посвящена описанию стандартных инструментальных средств и наилучшей практики организации совместной разработки программ на языке Python.

Глава 8. Производство

В языке Python предусмотрены средства адаптации к широкому кругу сред развертывания. Он также включает встроенные модули, нацеленные на повышение надежности и защищенности программ. В этой главе обсуждаются предлагаемые в Python средства отладки, оптимизации и тестирования, предназначенные для улучшения качества и производительности программ.

Соглашения, принятые в книге

Исходный код примеров, а также имена файлов, классов, объектов, методов и переменных выделяются в тексте моноширинным шрифтом. В необходимых случаях для разбивки длинных строк кода или текста используется символ ¶. Пропускам в коде примеров, обозначенным символами комментария (`#...`), соответствуют фрагменты кода, несущественные для текущего обсуждения. С целью сокращения размера приводимого кода описание примеров средствами встроенного документирования не применялось. Однако я настоятельно рекомендую вам использовать эту возможность при разработке собственных проектов, следуя общепринятым рекомендациям по стилю (раздел “Рекомендация 2. Руководствуйтесь правилами стилевого оформления программ, изложенными в документе PEP8”) и обязательно дополняя свой

код строками документирования (раздел “Рекомендация 49. Снабжайте строками документирования каждую функцию, класс и модуль”).

Для большинства примеров книги показан соответствующий вывод. Говоря о “выводе”, я подразумеваю выводимую на консоль или терминал информацию, которую вы видите при выполнении программы в окне интерпретатора команд. Разделы вывода выделяются моноширинным шрифтом, и им предшествует строка `>>>` (приглашение интерпретатора Python). Идея состоит в том, чтобы вы могли вводить код примеров в интерактивной оболочке Python и сравнивать полученные результаты с приведенными в книге.

Наконец, перед некоторыми разделами, выделенными моноширинным шрифтом, отсутствуют строки `>>>`. Такие разделы представляют вывод, полученный при выполнении программы вне интерактивной оболочки Python. Примеры этого типа часто начинаются символом `$`, указывающим на то, что данная программа запускалась в командной строке оболочки типа Bash.

Где найти код примеров

Некоторые примеры вам будет полезно просмотреть целиком, а не в виде фрагментов кода, перемежающихся с текстом. Это также даст вам возможность самостоятельно поработать с кодом, чтобы понять, почему программа работает именно так, а не иначе. Исходный код всех примеров доступен на сайте <http://www.effectivepython.com>. На этом же сайте вы найдете информацию об ошибках и опечатках, обнаруженных в книге.

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете при-слать нам бумажное или электронное письмо либо просто посетить наш сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интерес-ными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно оз-накомимся с вашим мнением и обязательно учтем его при отборе и под-готовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

в Украине: 03150, Киев, а/я 152