

ЙОАКИМ СУНДНЕС

PYTHON

АНАЛИЗ ДАННЫХ,
РАСЧЕТЫ И МОДЕЛИРОВАНИЕ

Издательство АСТ
г. Москва

УДК 004.43
ББК 32.973.2-018.1
С89

Книга «Introduction to Scientific Programming with Python» на английском языке, вынутая компаниями Springer и Simula в серию «Simula SpringerBriefs on Computing» (<http://www.springer.com/series/13548>), является изданием с открытым доступом и распространяется на условиях лицензии Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). Эта открытая книга доступна для свободного скачивания по ссылке: <https://doi.org/10.1007/978-3-030-50356-7>. В оформлении обложки использована иллюстрация из коллекции Shutterstock.com

Сунднес, Йоаким.

С89 Python: анализ данных, расчеты и моделирование : прикладное руководство / Йоаким Сунднес. — Москва : Издательство АСТ, 2026. — 224 с. : ил. — (Основы программирования. Прокачай навык).

ISBN 978-5-17-186112-4.

Перед вами подробное руководство по применению в научных целях Python — современного и наиболее востребованного языка программирования. Его преимущество заключается в том, что это так называемый язык высокого уровня с простым и интуитивно понятным синтаксисом. Однако несмотря на то, что Python хорошо подходит в качестве языка для начинающих, он также активно применяется и для решения самых сложных задач, благодаря чему в настоящее время является одним из наиболее широко используемых языков программирования во всем мире. Автор книги, норвежский исследователь Йоаким Сунднес, известный своими достижениями в области научных вычислений, машинного обучения и моделирования, подробно объясняет принципы работы с Python. Начиная с базового синтаксиса и заканчивая более сложными аспектами, в том числе применением инструментария объектно-ориентированного программирования на практике, он попутно знакомит читателя с использованием библиотек NumPy и Matplotlib для научных вычислений и визуализации научных данных.

Издание будет полезно специалисту с любым опытом и уровнем знаний. Изложение материала простое и доступное, что делает данное пособие отличным выбором для самостоятельного изучения. Оно прекрасно подойдет как для студентов, делающих первые шаги в программировании, так и для профессионалов, которые хотят овладеть новым инструментарием и внедрить Python в свои проекты. Читатели найдут здесь множество примеров решения практических задач, дополненных подробными объяснениями и комментариями.

УДК 004.43
ББК 32.973.2-018.1

© Joakim Sundnes, 2020

Перевод на русский язык: ООО «Интеджер».

ISBN 978-5-17-186112-4 Издание на русском языке: ООО «Издательство АСТ».

Содержание

Введение	7
Глава 1. Начало работы с Python	11
1.1. Первый пример: Hello, World!	11
1.2. Различные способы использования Python	12
Глава 2. Вычисления с использованием формул	16
2.1. Программирование простых математических вычислений	16
2.2. Переменные и типы переменных	18
2.3. Форматирование текстового вывода	24
2.4. Импорт модулей	27
2.5. Подводные камни при математическом программировании	30
Глава 3. Циклы и списки	34
3.1. Циклы для автоматизации повторяющихся задач	34
3.2. Булевы выражения	38
3.3. Использование списков для хранения последовательностей данных	40
3.4. Итерация по списку с помощью цикла for	42
3.5. Вложенные списки и нарезка списков	52
3.6. Кортежи	55
Глава 4. Функции и ветвление	58
4.1. Программирование с использованием функций	58
4.2. Аргументы функции и локальные переменные	62
4.3. Аргументы по умолчанию и дос-строки	71
4.4. If-проверки для ветвления потока программы	74
4.5. Функции как аргументы функций	78
4.6. Решение уравнений с помощью функций Python	80
4.7. Написание тестовых функций для проверки программ	85
Глава 5. Пользовательский ввод и обработка ошибок	92
5.1. Чтение данных пользовательского ввода	93
5.2. Гибкий пользовательский ввод с помощью eval и exec	98

5.3. Чтение данных из файлов	104
5.4. Запись данных в файлы	111
5.5. Обработка ошибок в программах	113
5.6. Создание модулей	120
Глава 6. Массивы и построение графиков	127
6.1. NumPy и вычисления с массивами	127
6.2. Построение кривых с помощью Matplotlib	135
6.3. Построение графиков непрерывных и кусочно-определенных функций.	141
6.4. Создание анимированного графика.	145
6.5. Другие полезные операции с массивами	153
Глава 7. Словари и строки.	157
7.1. Словари	157
7.2. Пример: словарь для многочленов.	163
7.3. Пример: чтение данных файла в словарь.	165
7.4. Операции со строками.	166
Глава 8. Классы.	176
8.1. Основы классов	176
8.2. Защищенные атрибуты класса.	185
8.3. Специальные методы.	188
8.4. Пример: автоматическое дифференцирование функций.	195
8.5. Тестовые функции для классов	197
8.6. Пример: класс многочленов	199
Глава 9. Объектно-ориентированное программирование.	205
9.1. Иерархия классов и наследование	205
9.2. Пример: классы для численного дифференцирования.	211
9.3. Пример: классы для численного интегрирования.	216
Указатель	221

*Посвящается памяти
Ханса Петтера Лангтангена.*

Введение

Эта книга изначально была написана как набор конспектов лекций к фундаментальному изданию «Введение в научное программирование на Python» (*A Primer on Scientific Programming with Python*) Ханса Петтера Лангтангена¹ и может быть использована либо как дополнение к нему, либо самостоятельно, в качестве комплексного введения в научное программирование. Книга Лангтангена и эти конспекты лекций составили основу вводного курса по научному программированию в Университете Осло. Курс ведется с 2007 года, его посещают в основном студенты первого курса математического, инженерного, физического, химического и геонаучного факультетов.

Написание этих конспектов лекций и их последующее превращение в книгу было продиктовано прежде всего двумя факторами. Первый заключался в том, что многие студенты находили почти 1000 страниц книги Лангтангена немного «подавляющими» в качестве начального введения в программирование. Этот эффект был в основном психологическим, поскольку книга хорошо структурирована и подходит для выборочного изучения глав и разделов, но отзывы студентов все равно указывали на необходимость более компактного и (в буквальном смысле) легкого введения. Вторым фактором стало то, что, к сожалению, Ханс Петтер Лангтанген скончался в 2016 году, и поэтому его книга не была обновлена примерами применения новейших версий Python. Эта проблема — также в основном ментальное препятствие, поскольку различия между версиями Python довольно малы, и требуются лишь незначитель-

¹ Hans Petter Langtangen, *A Primer on Scientific Programming with Python*, 5th edition, Springer-Verlag, 2016.

ные изменения кода, чтобы большинство примеров из оригинальной книги работали на новейшей платформе Python. Однако книга предназначена для введения в программирование, а при изучении совершенно новой темы любое незначительное несоответствие является потенциальным источником путаницы. Поэтому я посчитал необходимым создать обновленный материал, в котором все примеры кода будут работать без каких-либо изменений на наиболее распространенных платформах Python. Тем не менее, несмотря на эти незначительные недостатки, книга Лангтангена как вводный текст по-прежнему является отличным пособием по научному программированию на Python. По сравнению с настоящей книгой она охватывает гораздо более широкий круг тем и включает в себя больше примеров, более подробные обсуждения и объяснения, а также множество полезных советов и подсказок по программированию. Я настоятельно рекомендую его в качестве дополнения к этим заметкам всем, кто стремится стать экспертом в научном программировании.

Настоящая книга была написана специально для курса «Введение в программирование для научных приложений» (IN1900) в Университете Осло. Она в точности повторяет философию преподавания и общую структуру оригинальной книги Лангтангена, с главной идеей, что единственный способ научиться программировать — это писать программы. Читать теорию полезно, но без реальной практики программирования польза от нее очень мала. Таким образом, данный курс в значительной степени основан на решении задач и упражнениях по программированию, и основная цель этой книги — подготовить студентов к решению таких задач, предоставив краткое введение в фундаментальные концепции программирования и инструменты Python. Стил изложения компактный и прагматичный, включает большое количество примеров кода, иллюстрирующих работу и применение новых концепций на практике. Примеры представляют собой комбинацию фрагментов кода (так называемых снippetов), полных программ на Python и интерактивных сессий в оболочке Python. Читателям предлагается запускать и изменять код, чтобы понять, как работа-

ют различные концепции программирования. Исходный код большинства примеров, а также блокноты Jupyter для всех глав представлены в онлайн-ресурсах, сопровождающих эту книгу.

Для студента, изучающего математику, физику, химию или другие естественные науки, многие примеры из этой книги будут знакомы. Однако быстро растущая актуальность науки о данных означает, что вычисления и научное программирование будут представлять интерес для все большего числа пользователей. В этой книге не представлены типичные инструменты науки о данных, но читатель научится выполнять такие задачи, как считывание данных из файлов, простая обработка текста, программирование с использованием математики и вычислений с плавающей точкой. Все это — фундаментальные строительные блоки любого приложения для науки о данных, и их необходимо знать, прежде чем переходить к более продвинутым и специализированным инструментам.

Для чтения этой книги не требуется предварительных знаний о программировании. Мы стартуем с очень простых примеров, чтобы просто начать программировать, а затем перейдем к основам фундаментальных концепций программирования, таких как циклы, функции, if-проверки, списки и классы. Эти общие понятия дополняются более специфичными инструментами для научного программирования, в первую очередь для построения графиков и вычислений на основе массивов. Общая цель книги — познакомить читателя с программированием и, в частности, продемонстрировать, как оно может стать чрезвычайно полезным и мощным инструментом во всех отраслях естественных наук.

Многие люди внесли свой вклад в создание этой книги, в частности мои коллеги из исследовательской лаборатории Simula и Университета Осло. Однако вклад профессора Ханса Петтера Лангтангена на голову выше всех остальных. Профессор был чрезвычайно вдохновляющим наставником, ментором и коллегой на протяжении всей моей научной карьеры; он разработал курс, который сейчас известен как IN1900 в Университете Осло; он написал книгу, на которой основаны представленные здесь материалы. В этих

конспектах лекций я полностью скопировал идеи, стиль изложения и примеры кода из его оригинальной книги просто потому, что считаю их превосходными для знакомства с программированием в научном контексте. Если бы не Ханс Петтер, я бы никогда не написал этой книги и, возможно, я и не брался бы за нее, если бы профессор, к сожалению, не скончался в 2016 году — в этом труде не было бы необходимости, потому что он наверняка сам написал бы гораздо лучшую и более объемную книгу.

Май 2020 г.

Йоаким Сунднес

Глава 1.

Начало работы с Python

Эта книга посвящена изучению языка программирования Python, который является одним из самых популярных языков для вводных курсов по программированию. Преимущество Python заключается в том, что это так называемый язык высокого уровня, с простым и интуитивно понятным синтаксисом, который позволяет сразу же приступить к работе. Однако, несмотря на то, что Python хорошо подходит в качестве языка для начинающих, он также используется для решения более сложных задач и в настоящее время является одним из наиболее широко используемых языков программирования во всем мире.

1.1. Первый пример: Hello, World!

Большинство книг для новичков в программировании начинаются с простейшей программы «Hello, World!», которая просто выводит на экран «*Hello, World!*». В Python эта программа состоит всего из одной строки:

```
print("Hello, World!")
```

Чтобы написать и запустить такую программу, Python предлагает несколько различных вариантов. Во всей этой книге мы будем применять в основном классический подход к программированию, когда программа пишется в текстовом редакторе и сохраняется в виде файла, который затем запускается из окна командной строки или из интегрированной среды разработки (IDE). Чтобы написать и запустить приведенную выше программу «Hello, World!», откройте ваш любимый редактор (Atom, gedit, Emacs и т. д.), наберите указанную строку и сохраните файл с под-

ходящим именем например `hello.py`¹. Затем откройте терминал или окно iPython, перейдите в каталог, где вы сохранили файл, и введите `python hello.py`, если вы используете обычный терминал, или `run hello.py`, если вы используете iPython. В результате `Hello, World!` должно появиться в терминале сразу после ввода команды. Если вы используете IDE, то это, по сути, редактор и окно iPython/терминала, объединенные вместе. Например, в популярной IDE Spyder редактор обычно находится в левом верхнем окне, а окно в правом нижнем углу — это окно iPython, в котором вы запускаете программу².

Хотя программа «Hello, World!» может показаться глупым примером, она служит ряду полезных целей. Прежде всего запуск этой небольшой программы позволяет убедиться в том, что вы правильно установили Python, а также в том, что вы установили нужную версию. Она также знакомит с функцией `print`, которая будет использоваться практически каждый раз, когда мы будем программировать, и иллюстрирует, как мы используем кавычки для определения *строки* в Python. Если `print` — это слово, которое Python понимает, то слова «Hello» и «World» — нет. Используя кавычки, мы сообщаем Python, что он не должен пытаться понять (или интерпретировать) эти слова, а скорее рассматривать их как простой текст, который в данном случае должен быть выведен на экран. Более подробно мы вернемся к этой теме позже.

1.2. Различные способы использования Python

Как уже было сказано выше, Python предлагает несколько альтернатив традиционному стилю программирования с использованием текстового редактора и окна терминала, и некоторые из этих альтернатив могут быть очень полезны при обучении программированию. Например, мы можем ис-

¹ Мы не описываем здесь технические детали обретения и установки Python, поскольку эта информация зависит от платформы и очень быстро устаревает. Актуальные советы по установке Python можно найти на веб-странице курса IN1900 в Университете Осло (<https://www.uio.no/studier/emner/matnat/ifi/IN1900/index-eng.html>) или на других многочисленных ресурсах, доступных в сети.

² Подробнее см., например, <https://www.spyder-ide.org/>.

пользовать Python в интерактивном режиме, просто набрав `python` или `ipython` в окне терминала, без последующего указания имени файла. Это откроет среду для ввода и выполнения команд Python, которая не очень подходит для написания программ в несколько строк, но весьма полезна для тестирования команд и операторов Python или просто для использования Python в качестве калькулятора. В обычном окне терминала на macOS или Linux интерактивная версия примера «Hello, World!» будет выглядеть примерно так:

```
Terminal> ipython
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.4.0 -- An enhanced Interactive Python.
In [1]: print("Hello, World!") Hello, World!
```

In [2]:

Отметим, что `python` и `ipython` работают практически одинаково, но `ipython` имеет ряд дополнительных возможностей и рекомендуется к использованию.

Третий способ использования Python — *блокноты Jupyter (Jupyter notebooks)*, которые представляют собой разновидность интерактивных блокнотов, объединяющих код и текст. Блокноты просматриваются через браузер и внешне очень похожи на обычную веб-страницу, но с тем важным отличием, что сегменты кода — это «живой» код Python, который можно запускать, изменять и перезапускать во время чтения документа. Эти возможности особенно полезны для целей обучения, поскольку подробные объяснения новых концепций легко сочетаются с интерактивными примерами. Все главы этой книги также доступны в виде блокнотов Jupyter.

Небольшие недостатки языка Python. Несмотря на то, что Python — очень популярный и подходящий язык для обучения программированию, у него есть и некоторые незначительные недостатки. Один из самых важных тесно свя-

зан с его преимуществом: с тем, что Python — язык высокого уровня с коротким и интуитивно понятным синтаксисом. Написание небольших программ на Python может быть очень эффективным, и новички могут быстро начать писать полезные программы, но недостатком является то, что код может становиться беспорядочным по мере роста размера и сложности программ. Другие языки, такие как C, C++ и Java, требуют большей структурированности кода, что может смущать новичков и раздражать, когда нужно быстро написать небольшую программу, но в долгосрочной перспективе это повышает эффективность при написании больших программ. Однако, безусловно, и на Python можно писать аккуратные и хорошо структурированные программы, но это требует от программиста следования определенным принципам стиля кодирования.

Еще один немного раздражающий аспект Python — это то, что он существует в разных версиях. На момент написания этой книги Python 3 был доминирующим уже довольно долгое время, но если вы поищете ресурсы по программированию в Интернете или прочитаете старые учебники, вы найдете множество примеров, использующих Python 2. Для программирования, ориентированного на математику и рассматриваемого в этой книге, разница между Python 2 и Python 3 на самом деле довольно мала, но некоторые различия важны. Самое очевидное из них — это принцип работы той же функции (в Python 2 — оператора) `print`. В Python 2 приведенная выше программа будет считывать `print "Hello, World!"`, то есть без скобок. Поскольку почти во всех примерах кода в той или иной степени используется `print`, программы, написанные на Python 2, как правило, не будут работать в Python 3.

Одним из особенно актуальных ресурсов по научному использованию Python является «Введение в научное программирование на Python» Ханса Петтера Лангтангена¹ (на этом учебнике во многом основана данная книга). Однако последняя версия

¹ Hans Petter Langtangen, A Primer on Scientific Programming with Python, 5th edition, Springer-Verlag, 2016.

этой книги была издана в 2016 году, и все примеры кода написаны на Python 2, соответственно, при их запуске на Python 3 результатом будет сообщение об ошибке. В большинстве случаев единственной ошибкой является отсутствие круглых скобок; поэтому добавление круглых скобок ко всем `print` сделает большинство примеров нормально работающими в Python 3. О других различиях между версиями Python мы расскажем позже.