

004  
У-497

**ВЫСШЕЕ**

**ОБРАЗОВАНИЕ**

С. А. Чернышев

# ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON

2-е издание



Курс с on-line  
оцениванием

**УМО ВО**  
РЕКОМЕНДУЕТ

 **юрайт**  
издательство

С. А. Чернышев

# ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON

УЧЕБНОЕ ПОСОБИЕ ДЛЯ ВУЗОВ

2-е издание, переработанное и дополненное

*Рекомендовано Учебно-методическим отделом высшего образования в качестве учебного пособия для студентов высших учебных заведений, обучающихся по IT-направлениям*



Курс с практическими заданиями и дополнительными материалами доступен на образовательной платформе «Юрайт», а также в мобильном приложении «Юрайт.Библиотека»

Москва • Юрайт • 2024

УДК 004.432(075.8)

ББК 32.973.2я73

Ч-49

**Автор:**

**Чернышев Станислав Андреевич** — кандидат технических наук, доцент кафедры интеллектуальных систем и защиты информации Института информационных технологий и автоматизации Санкт-Петербургского государственного университета промышленных технологий и дизайна, доцент кафедры информатики факультета информатики и прикладной математики Санкт-Петербургского государственного экономического университета.

**Рецензенты:**

**Гордеев А. В.** — профессор, доктор технических наук, профессор кафедры вычислительных систем и сетей Института вычислительных систем и программирования Санкт-Петербургского государственного университета аэрокосмического приборостроения;

**Штеренберг С. И.** — кандидат технических наук, доцент кафедры защищенных систем связи факультета инфокоммуникационных сетей и систем Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича.

**Чернышев, С. А.**

Ч-49 Основы программирования на Python: учебное пособие для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 349 с. — (Высшее образование). — Текст : непосредственный.

ISBN 978-5-534-17139-6

В курсе подробно описывается не только большое количество базовых понятий и операторов языка программирования Python, но и ряд нюансов, с которыми так или иначе предстоит встретиться при его использовании в процессе написания программных продуктов. Материал подается по принципу «от простого к сложному» и сопровождается большим количеством примеров и упражнений, что позволяет сформировать у студентов практические навыки программирования и тестирования разрабатываемых приложений. Все исходные коды рассматриваемых примеров можно скачать с репозитория автора на GitHub.

Соответствует актуальным требованиям федерального государственного образовательного стандарта высшего образования.

Курс предназначен для студентов высших учебных заведений, которые обучаются по инженерно-техническим направлениям.

УДК 004.432(075.8)

ББК 32.973.2я73

553.839



Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

ISBN 978-5-534-17139-6

© Чернышев С. А., 2021

© Чернышев С. А., 2023, с изменениями

© ООО «Издательство Юрайт», 2024

# Оглавление

<b>Введение.....</b>	<b>9</b>
<b>Тема 1. Краткая история Python, его особенности и типы данных.....</b>	<b>13</b>
1.1. Краткая история языка программирования Python.....	13
1.2. В Python все является объектом.....	17
1.3. В Python нет переменных.....	18
1.4. Интернированные (intern) объекты в Python .....	26
1.5. Глобальная блокировка интерпретатора.....	29
1.6. Подходы к сборке мусора в Python.....	32
1.6.1. Алгоритм подсчета ссылок.....	33
1.6.2. Garbage Collector (GC).....	33
1.6.3. Слабые ссылки .....	35
1.7. Встроенные типы данных Python.....	37
1.7.1. Строки.....	38
1.7.2. Списки .....	41
1.7.3. Словари .....	45
1.7.4. Кортежи.....	49
1.7.5. Файлы .....	51
1.7.6. Множества.....	56
Резюме.....	60
Вопросы и задания для самопроверки.....	60
Упражнения.....	61
<b>Тема 2. Синтаксис, операторы и управляющие конструкции... 64</b>	<b>64</b>
2.1. Основные операторы в Python .....	64
2.2. Использование отступов в Python .....	70
2.3. Комментарии.....	71
2.4. Правила именования переменных (имен).....	71
2.5. Оператор if и match .....	72
2.6. Цикл while .....	78
2.6.1. Работа цикла с операторами break, continue, pass .....	80
2.7. Цикл for.....	82
2.8. Различные способы написания циклов .....	84
2.8.1. Использование встроенной функции range .....	84
2.8.2. Использование встроенной функции zip.....	86
2.8.3. Использование встроенной функции enumerate .....	87

2.9. Итерации и включения .....	88
2.9.1. Итератор и итерируемый объект .....	88
2.9.2. Списковые включения (list comprehension) .....	89
2.10. Источники документации Python .....	91
Резюме .....	97
Вопросы для самопроверки .....	98
Упражнения .....	99
<b>Тема 3. Функции в Python.....</b>	<b>100</b>
3.1. Области видимости .....	102
3.1.1. Области видимости и вложенные функции .....	103
3.1.2. Пример работы с областями видимости .....	104
3.1.3. Замыкания.....	105
3.2. Аргументы функции .....	107
3.2.1. Значения аргументов по умолчанию .....	110
3.2.2. Режимы сопоставления аргументов функции .....	111
3.3. Возвращение результатов выполнения функцией .....	113
3.4. Рекурсия .....	115
3.5. Аннотация функций .....	116
3.6. Лямбда-функции (выражения) .....	118
3.7. Декораторы.....	119
3.8. Генераторы .....	123
3.8.1. Генераторные функции.....	123
3.8.2. Генераторные выражения .....	126
Резюме .....	127
Вопросы и задания для самопроверки.....	127
Упражнения .....	129
<b>Тема 4. Модули и пакеты.....</b>	<b>131</b>
4.1. Принцип работы импортирования.....	132
4.1.1. Поиск файла подключаемого модуля .....	132
4.1.2. Компиляция импортируемого модуля в байт-код.....	134
4.1.3. Выполнение кода модуля .....	135
4.2. Создание и использование модулей .....	135
4.2.1. Пространства имен модулей.....	138
4.2.2. Перезагрузка модулей .....	140
4.2.3. Запуск модуля как автономной программы .....	141
4.2.4. Внутренние имена в модулях .....	142
4.3. Создание и использование пакетов модулей .....	144
Резюме .....	147
Вопросы и задания для самопроверки.....	147
Упражнения .....	148
<b>Тема 5. Классы и объектно-ориентированное программирование.....</b>	<b>149</b>
5.1. Определение класса .....	149
5.2. Имена (переменные) экземпляров класса .....	151

5.3. Методы экземпляра класса .....	151
5.4. Имена (переменные) класса .....	152
5.5. Статические методы.....	154
5.6. Методы класса .....	154
5.7. Приватные методы и переменные .....	155
5.8. Наследование .....	156
5.9. Множественное наследование .....	161
5.10. Абстрактные классы и переопределение методов .....	163
5.11. Перегрузка операций .....	165
5.11.1. Перегрузка <code>__add__</code> , <code>__or__</code> , <code>__sub__</code> .....	167
5.11.2. Перегрузка <code>__getitem__</code> и <code>__setitem__</code> .....	167
5.11.3. Перегрузка <code>__iter__</code> и <code>__next__</code> .....	169
5.11.4. Перегрузка <code>__contains__</code> .....	173
5.11.5. Перегрузка <code>__getattr__</code> и <code>__setattr__</code> .....	173
5.11.6. Перегрузка <code>__repr__</code> и <code>__str__</code> .....	175
5.11.7. Перегрузка <code>__call__</code> .....	176
5.11.8. Перегрузка методов сравнения .....	178
5.11.9. Перегрузка <code>__len__</code> и <code>__bool__</code> .....	178
5.11.10. Декоратор <code>@property</code> или свойства .....	179
5.12. Вложенные классы и пространство имен.....	180
5.13. Перечисления (Enum) .....	182
5.13.1. <code>IntEnum</code> .....	185
5.13.2. <code>IntFlag</code> .....	186
5.13.3. <code>Flag</code> .....	186
5.13.4. Расширенные возможности перечислений.....	187
Резюме .....	188
Вопросы для самопроверки.....	189
Упражнения .....	189
<b>Тема 6. Исключения (Exception) .....</b>	<b>191</b>
6.1. Пользовательские исключения.....	192
6.2. Основы обработки и генерации исключений .....	193
6.2.1. Унифицированный оператор <code>try/except/finally</code> .....	194
6.2.2. Оператор <code>raise</code> .....	197
6.2.3. Оператор <code>assert</code> .....	199
6.2.4. Оператор <code>with/as</code> и протокол управления контекстами .....	199
6.3. Встроенные классы исключений .....	201
Резюме .....	202
Вопросы и задания для самопроверки.....	203
Упражнения .....	203
<b>Тема 7. Потоки, процессы и асинхронное программирование.....</b>	<b>205</b>
7.1. Многопоточное программирование .....	205
7.1.1. Модуль <code>threading</code> и класс <code>Thread</code> .....	206
7.1.2. Потоки <code>Timer</code> .....	207

7.1.3. Класс RLock. Управление доступом к ресурсам .....	208
7.1.4. Синхронизация потоков .....	210
7.1.5. Семафоры .....	212
7.2. Multiprocessing .....	214
7.2.1. Модуль multiprocessing и класс Process .....	214
7.2.2. Взаимодействие между процессами .....	215
7.2.3. Пул процессов .....	218
7.3. Асинхронное программирование .....	219
7.3.1. Планирование времени вызова функций .....	221
7.3.2. Асинхронное получение результатов .....	222
7.3.3. Параллельное выполнение задач .....	223
7.4. Что и когда использовать? .....	224
Резюме .....	224
Вопросы и задания для самопроверки .....	225
Упражнения .....	225
<b>Тема 8. Разработка графического пользовательского интерфейса .....</b>	<b>227</b>
8.1. Установка PySide2 .....	227
8.2. Основы разработки GUI .....	228
8.2.1. Hello World .....	228
8.2.2. QPushButton и QLabel .....	229
8.2.3. QLineEdit .....	230
8.2.4. QCheckBox .....	232
8.2.5. Компоновка элементов GUI .....	234
8.3. Пользовательские виджеты и сигнал-слотовый механизм .....	237
8.4. Использование Qt Designer для разработки GUI .....	242
Резюме .....	246
Вопросы и задания для самопроверки .....	247
Упражнения .....	247
<b>Тема 9. Сетевое программирование .....</b>	<b>249</b>
9.1. Архитектура «клиент-сервер» .....	251
9.1.1. Логика работы без установления соединения .....	251
9.1.2. Логика работы с установлением соединения .....	252
9.2. Модуль socket .....	253
9.3. Пример клиента и сервера, работающих без установления соединения .....	254
9.4. Пример клиента и сервера, работающих с установлением соединения .....	256
9.5. Фреймворк для сетевых серверов — socketserver .....	258
9.5.1. TCP эхо-сервер с использованием TCPServer .....	258
9.5.2. UDP эхо-сервер с использованием UDPServer .....	259
Резюме .....	260
Вопросы и задания для самопроверки .....	260
Упражнения .....	260



<b>Тема 10. Хранение данных и обмен данными .....</b>	<b>262</b>
10.1. Создание базы данных SQLite .....	262
10.2. Использование переменных в запросах .....	266
10.2.1. Запросы с позиционными параметрами .....	266
10.2.2. Запросы с именованными параметрами.....	267
10.3. Транзакции .....	267
10.4. Уровни изоляции (доступа) .....	271
Резюме .....	274
Вопросы для самопроверки.....	274
Упражнения .....	275
<b>Тема 11. Тестирование .....</b>	<b>276</b>
11.1. Тестирование с использованием библиотеки PyTest.....	277
11.1.1. Простые тесты .....	277
11.1.2. Запуск одного или нескольких тестов .....	279
11.1.3. Запуск подмножества тестов .....	280
11.1.4. Использование фикстур.....	282
11.1.5. Параллельный запуск тестов .....	285
11.2. Тестирование с использованием unittest.....	285
11.2.1. Пример простого теста .....	285
11.2.2. Использование объектов-подделок .....	286
Резюме .....	287
Вопросы и задания для самопроверки.....	287
Упражнения .....	287
<b>Тема 12. Разработка telegram-бота .....</b>	<b>288</b>
12.1. Настройка окружения проекта .....	288
12.2. Загрузка первичных данных из файла MS Excel.....	290
12.3. Создание моделей.....	292
12.4. Инициализация базы данных .....	296
12.5. Разработка telegram-бота.....	299
12.5.1. Разработка функциональности для запуска бота .....	300
12.5.2. Отмечаем отсутствующих .....	306
12.5.3. Добавляем дисциплину, группу и студентов.....	315
12.5.4. Удаление студента.....	320
12.5.5. Назначение дисциплины группе .....	323
12.5.6. Генерация интерактивного отчета.....	325
12.5.7. Генерация отчетов в Excel .....	330
Резюме .....	343
Вопросы и задания для самопроверки.....	343
Упражнения .....	343
<b>Список используемых источников .....</b>	<b>345</b>
<b>Новые издания по дисциплине «Программирование</b>	
<b>на языке Python» и смежным дисциплинам .....</b>	<b>349</b>



## Введение

Данная книга познакомит вас с основами языка программирования Python. Конечно, в одной книге невозможно рассмотреть все аспекты и нюансы этого языка программирования, но автор проделал значительную работу, стараясь максимально осветить его базовые возможности.

Все исходные коды приведенных в книге примеров можно скачать из репозитория автора: [https://github.com/MADTeacher/python\\_basics](https://github.com/MADTeacher/python_basics). Для их запуска вам понадобится open-source экосистема Anaconda с версией Python 3.10 или старше<sup>1</sup>, которая включает в себя такой инструмент, как Jupyter Notebook, а также интегрированная среда разработки PyCharm<sup>2</sup>. Их установка, настройка и использование не представляют сложности, поэтому и не рассматриваются в данной книге, но если у вас возникнут какие-то трудности, то в сети интернет достаточно материала для их решения.

Это сделано специально, так как в процессе работы в любой организации (да даже над упражнениями данной книги) вам все равно, так или иначе, придется осуществлять поиск нужной информации в интернете. Неважно, какую позицию в дальнейшем придется занимать, от умения быстро найти ту информацию, которая позволит решить возникающие проблемы, будет зависеть очень многое. И именно в этом месте кроется первая ловушка, в которую попадают начинающие разработчики — они бездумно копируют найденный код и переносят его в рабочий проект. Пусть на начальных стадиях это и поможет двигаться быстрыми шагами, но вы даже не представляете, какую бомбу замедленного действия подкладываете всем участникам проекта. Конечно, есть code review, одной из задач которого является поиск таких моментов. Но все мы люди и имеем свойство ошибаться. Особенно, когда трещат сроки проекта и приходится постоянно задерживаться на работе, чтобы успеть в обозначенный договором срок. К тому же в такие моменты зачастую не до code review. И мало того, если сборка проекта будет падать с указанием ошибок на ваш кусок кода, вы не сможете ответить на вопрос: «Что этот код в действительности делает?» (а не ка-

---

<sup>1</sup> Она доступна для скачивания по ссылке: <https://www.anaconda.com/products/individual>.

<sup>2</sup> <https://www.jetbrains.com/ru-ru/pycharm/download/>

кую вашу проблему он решил). Поэтому, если хотите перенести найденный код в проект, не копируйте его, а перепечатайте вручную. Да, это медленнее, но это позволит вам иметь представление, что же на самом деле происходит (или найти ошибки), а не верить на слово тому человеку, который выложил этот код в открытый доступ.

Теперь коснемся упражнений к разделам. Упражнения к темам с 1 по 7 предназначены для выполнения одним человеком. Вы и далее, конечно, можете продолжать выполнять упражнения в одиночку, но лучше скооперироваться с кем-нибудь и выполнять их, как минимум, в паре. Этот опыт вам пригодится в дальнейшем, поскольку разработка современных информационных систем ведется командой, а не одиночками. Таким образом, работая в паре, вы научитесь:

- 1) декомпонировать задачи, чтобы их можно было выполнять параллельно, а после интегрировать в одно приложение;

- 2) работать в команде, что подразумевает постоянное обсуждение возникающих трудностей и нахождение компромисса относительно их решения. Так, например, для начала попробуйте договориться о стиле кодирования, которого будете придерживаться (или возьмите за основу PEP 8);

- 3) тайм-менеджменту, а именно предварительной оценке необходимого времени для решения задач и упражнения в целом. Не переживайте, если сначала не будет получаться. Все дело в опыте (и ваша задача — его получить).

Само собой, это еще не весь список того, какой опыт вы получите при выполнении упражнений с кем-то, а не в одиночку, так как с собой договориться намного проще, чем с другим человеком.

В первой теме книги затрагиваются такие моменты, как: встроенные переменные, менеджмент памяти, глобальная блокировка интерпретатора, интернированные объекты и зачем они нужны, а также почему принято считать, что в Python нет переменных.

Вторая тема посвящена синтаксису Python. В ней рассматривается, какие операторы существуют и как используются, правила именования переменных, уровни отступов, что такое ветвление и циклы, для чего используется протокол итерации, как он работает и т. д.

В третьей теме речь пойдет о функциях и всем, что с ними связано: областях видимости, замыканиях, аннотациях, генераторах, декораторах и т. д.

Четвертая тема приоткрывает завесу тайн над модулями и пакетами, описывает, как они импортируются и что при этом происходит, а также способы их создания.

В пятой теме рассматривается, что такое объектно-ориентированное программирование и то, как его концепции (инкапсуляция,

наследование и полиморфизм) реализованы в Python. Какие существуют различия между: классом и экземпляром класса, переменной (методом) класса и экземпляром класса и т. д.

В шестой теме затрагивается такой аспект всех современных языков программирования, как исключения, которые являются высокоуровневым механизмом управления потоком выполнения и могут генерироваться интерпретатором или самим программистом «вручную».

В седьмой теме обсуждаются многопоточное, многопроцессорное и асинхронное программирование средствами языка программирования Python, их различия и ситуации, в которых их следует использовать.

В восьмой теме рассматривается работа с библиотекой PySide2, которая позволяет разрабатывать приложения с графическим пользовательским интерфейсом и является официальной реализацией кроссплатформенного фреймворка Qt для Python, развитие которой поддерживается The Qt Company.

Девятая тема — сетевое программирование средствами Python, базовые протоколы обмена данными с установлением и без установления соединения между клиентами и сервером.

В десятой теме рассматривается работа с встроенной реляционной базой данных — SQLite, средствами модуля *sqlite3*.

В одиннадцатой теме рассматриваются способы тестирования разрабатываемых приложений на языке программирования Python и некоторые инструменты для этого.

В двенадцатой теме рассматривается разработка telegram-бота с поддержкой ORM SQLAlchemy.

Хочу выразить огромную благодарность свой невесте, ученикам (Антонову Александру, Быкову Алексею, Дуку Герману, Нагорных Максиму и Твардовскому Георгию) за оказанную помощь и поддержку в процессе написания данной книги.

В результате изучения материалов данного учебного пособия обучающиеся должны:

**знать**

- стандартные модули языка программирования Python;
- основные типы, структуры данных и особенности работы с ними;
- синтаксис языка программирования;
- ключевые особенности интерпретатора CPython;
- принципы разработки telegram-ботов;
- принципы использования ORM SQLAlchemy для работы с базами данных;

**уметь**

- применять стандартные структуры данных при разработке приложений;

- разрабатывать многопоточные, асинхронные и многопроцессорные приложения;
- разрабатывать приложения с графическим пользовательским интерфейсом;
- писать тесты для проверки правильности функционирования разрабатываемых приложений;
- разрабатывать приложения, устойчивые к возникающим ошибкам в процессе их работы;
- разрабатывать приложения, способные взаимодействовать между собой по сети;
- разрабатывать telegram-ботов,
- работать с базами данных посредством ORM SQLAlchemy;

**владеть**

- основными принципами объектно-ориентированного программирования;
- навыками разработки многопоточных, асинхронных и многопроцессорных приложений;
- навыками работы с исключениями;
- навыками разработки клиент-серверных приложений;
- навыками разработки telegram-ботов,
- основами тестирования приложений.