

УДК 004.43
ББК 32.973.2-018.1
В14

PYTHON 3 FÜR STUDIUM UND AUSBILDUNG
EINFACH LERNEN UND PROFESSIONELL ANWENDEN

Michael Weigend

© 2022 mitp Verlags GmbH & Co.KG

Published in German Language as: Python 3 für Studium und Ausbildung by Michael Weigend,
1st Edition 2022 by MITP Verlag, Germany / All Rights Reserved. Published with arrangements made
by Maria Pinto-Peuckmann, Literary Agency — World Copyright Promotion, Kaufering, Germany

Вайгенд, Михаэль.

В14 Python для учебы и работы. Как освоить самый мощный язык программирования для будущей карьеры / Михаэль Вайгенд ; [перевод с английского Е. Чернышовой]. — Москва : Эксмо, 2024. — 464 с. — (Мировой компьютерный бестселлер).

ISBN 978-5-04-191815-6

«Python для учебы и работы» — идеальное пособие для начинающих программистов. Книга предлагает простое и эффективное изучение одного из самых популярных языков программирования без необходимости иметь глубокие знания. Упражнения и практические примеры охватывают различные области применения, позволяя подготовиться к широкому спектру задач.

УДК 004.43
ББК 32.973.2-018.1

ISBN 978-5-04-191815-6

© Евгения Чернышова, перевод на русский язык, 2024
© Оформление. ООО «Издательство «Эксмо», 2024

Для Ларса Йона

Оглавление

Введение	17
Структура	17
Обратите внимание на шрифт!	18
Тексты программ и ключи для скачивания	19
Глава 1 Добро пожаловать в Python!	20
1.1 Язык программирования Python	20
1.2 Что такое алгоритм?	21
1.3 Синтаксис и семантика	21
1.4 Интерпретаторы и компиляторы	22
1.5 Установите Python	23
1.6 Python в интерактивном режиме	24
1.7 Среда разработки IDLE	26
1.8 Горячие клавиши оболочки IDLE	26
1.9 Команды	27
1.9.1 Выражения	27
1.9.2 Вызов функций	28
1.9.3 Присваивание значения	28
1.9.4 Расширенное присваивание	32
1.10 Работа с числами: Python в качестве калькулятора	32
1.10.1 Операторы	32
1.10.2 Использование переменных	34
1.11 Среда разработки Thonny	35
1.12 Блокноты с Jupyter и CoLab	37
1.13 Повторение	38
1.14 Упражнения	38
1.15 Решение вопроса: семантика в обычной жизни	40
Глава 2 Типы данных — иерархия типов в Python	41
2.1 Литералы и функция type()	41
2.2 Иерархия типов в Python	42
2.3 Стандартные типы	43
2.3.1 Целое число (int)	43
2.3.2 Числа с плавающей запятой (float)	44
2.3.3 Комплексные числа (complex)	45
2.3.4 Строки (str)	46
2.3.5 Кортежи (tuple)	46
2.3.6 Список (list)	47

2.3.7	Множество	47
2.3.8	Словарь	48
2.3.9	Логические значения — тип данных bool	48
2.3.10	NoneType	49
2.4	Общие операции с коллекциями	49
2.4.1	Коллекция	50
2.4.2	Последовательность	50
2.5	Создание объектов типа — приведение типов	52
2.6	Динамическая типизация	53
2.7	Повторение	53
2.8	Упражнение. Операторы	54
Глава 3	Интерактивные программы	55
3.1	Первый скрипт Python	55
3.2	Принцип работы «ввод — обработка — отображение»	58
3.3	Комментарии	60
3.4	Проект «Расчет объема»	60
3.4.1	Более короткий текст программы за счет вложенных вызовов функций	62
3.4.2	Вызов функции print() с несколькими аргументами	62
3.5	Запуск программ Python	63
3.5.1	Выполнение в командной строке	63
3.5.2	Запуск программы через значок программы в Windows ...	65
3.5.3	Программы Python в Linux — строка shebang	67
3.5.4	Запуск в macOS Finder	67
3.6	Поиск ошибок	67
3.6.1	Синтаксические ошибки	67
3.6.2	Ошибки выполнения	68
3.6.3	Семантические ошибки	69
3.6.4	Советы по устранению ошибок	69
3.7	Повторение	70
3.8	Упражнения	70
3.9	Ответы на вопросы	72
Глава 4	Структуры управления	74
4.1	Ветвления	74
4.1.1	Односторонняя ветвь (if)	74
4.1.2	Проект «Пароль»	75
4.1.3	Двустороннее ветвление (if...else)	76
4.1.4	Проект «Билет в кино»	77
4.1.5	Множественное ветвление (if...elif...else)	78
4.1.6	Проект «Информационная машина»	79

4.2	Компоновка программ Python: строки и блоки	80
4.2.1	Блок	80
4.2.2	Структура строки	81
4.3	Выстраивание условий	82
4.3.1	Булевы значения	82
4.3.2	Булевы операторы	83
4.3.3	Цепочки сравнений	83
4.3.4	Проект «Распознавание фруктов»	84
4.4	Цикл while	86
4.4.1	Проект «Суммирование»	87
4.4.2	Проект «Планеты»	88
4.4.3	Проект «Вычисление корня» (математика)	89
4.4.4	Бесконечные повторы	91
4.5	Итерации — for	92
4.5.1	Повторы с помощью функции range()	93
4.6	Повторение:	95
4.7	Упражнения	95
4.8	Ответы на вопросы	97
Глава 5	Функции	98
5.1	Для чего определяют функции?	98
5.2	Назначение и вызов функции	99
5.2.1	Проект «Расчет высоты падения»	99
5.3	Дополнительные параметры и значения по умолчанию	101
5.3.1	Расширение проекта: высота падения на различных небесных телах	101
5.4	Тестирование функции в оболочке	103
5.5	Оператор return	103
5.5.1	Процедуры	103
5.5.2	Эффекты оператора return	103
5.6	Позиционные аргументы и аргументы ключевых слов	104
5.7	Хороший стиль программирования	106
5.7.1	Название функции	106
5.7.2	Аннотации функций	106
5.7.3	Строка документации Docstring	107
5.7.4	Сигнатура	108
5.8	Функция print() более детально	108
5.9	Глобальные и локальные имена	109
5.10	Рекурсивные функции	110
5.10.1	Проект «Расчет факториала»	111
5.11	Лямбда-функции*	113
5.12	Функции в качестве аргументов map() и filter()*	114

5.12.1	Сопоставление	114
5.12.2	Фильтр	116
5.13	Повторение	117
5.14	Упражнения	118
5.15	Ответы на вопросы	119
Глава 6	Работа с модулями	121
6.1	Инструкции по импорту	121
6.1.1	Импорт модуля	121
6.1.2	Импорт функций из модуля	122
6.2	Математические функции: модуль <code>math</code>	123
6.3	Случайные функции: модуль <code>random</code>	124
6.3.1	Проект «Кости»	124
6.3.2	Проект «Кто следующий?»	125
6.4	Дата и время	125
6.4.1	Проект «Время»	127
6.4.2	Проект «Тренажер навыков счета»	127
6.5	Создание собственного модуля	129
6.5.1	Проект «Модуль для вычисления объема»	129
6.5.2	В чем преимущество модулей?	133
6.6	Модули из Python Package (PyPI)	133
6.7	Повторение	134
6.8	Упражнения	134
Глава 7	Модель с коллекциями	136
7.1	Последовательности	136
7.1.1	Списки	136
7.1.2	Кортеж	137
7.1.3	Сложные последовательности	137
7.1.4	Итерация по списку кортежей	138
7.1.5	Общие операции для последовательностей	139
7.1.6	Специальные операции для списков	141
7.1.7	Сортировка	142
7.1.8	Создание списка	143
7.2	Проект «Список телефонов»	144
7.3	Словари	147
7.3.1	Операции со словарями	148
7.3.2	Изменение словаря	148
7.4	Проект «Словарный тренажер»	149
7.5	Проект «Планировщик маршрута»	151
7.5.1	Маршруты и графы	151
7.5.2	Программирование	153

7.6	Повторение	156
7.7	Упражнения	156
7.8	Ответы на вопросы	157
Глава 8	Сохранение данных	159
8.1	Как хранятся данные?	159
8.1.1	Открытые файлы	159
8.1.2	Потоковые stream-методы	160
8.1.3	Сохранение и загрузка текстов	161
8.1.4	Двоичные файлы и строки байтов	162
8.1.5	Пути в дереве каталогов	162
8.2	Как избежать ошибок времени исполнения	164
8.2.1	try...except	164
8.2.2	try...except...finally	165
8.3	Оператор with	166
8.4	Проект «Журнал записей»	167
8.5	Сохранение и загрузка структур данных: модуль pickle	169
8.5.1	Сохранение	169
8.5.2	Загрузка	170
8.6	Проект «Цифровой планер»	171
8.7	Сохранение данных в формате JSON	174
8.7.1	Структура текста JSON	175
8.7.2	Ограничения JSON	176
8.8	Проект «Температурные данные»	177
8.8.1	Writer	177
8.8.2	Reader	178
8.9	Данные из Интернета	179
8.10	Проект «Электронная библиотека»	180
8.11	Повторение	182
8.12	Упражнение «Проверка новостей»	182
8.13	Ответы на вопросы	183
Глава 9	Обработка текста	184
9.1	Номера Unicode для символов	184
9.2	Escape-последовательности	185
9.3	Строковые методы	186
9.4	Проект «Словарь Гете»	188
9.5	Проект «Какая будет погода?»	190
9.6	Регулярные выражения*	192
9.6.1	Что такое регулярное выражение?	192
9.6.2	Структура регулярного выражения	193
9.6.3	Поиск фрагментов текста с помощью findall()	194

9.6.4	Заполнитель для символов из набора символов	197
9.6.5	Объединение регулярных выражений	198
9.6.6	Квантификаторы	198
9.6.7	Маска специальных символов	199
9.6.8	Жадный или нежадный поиск	200
9.6.9	Веб-скрапинг с использованием регулярных выражений	202
9.7	Тексты с переменными частями	203
9.7.1	Форматирование	203
9.7.2	Заполнители с именами	204
9.7.3	Спецификации формата для заполнителей	204
9.8	Проект «Анализ текста»	205
9.9	Проект «Сторителлинг»	206
9.10	Повторение	207
9.11	Упражнения	208
9.12	Ответы на вопросы	210
Глава 10	Доступ к системной среде	212
10.1	Интерфейс к операционной системе: модуль os	212
10.2	Поиск и определение свойств	213
10.2.1	Подкаталоги	214
10.2.2	Дерево каталогов	215
10.3	Создание и переименование файлов и каталогов	217
10.3.1	Проект «Переименование файлов изображений»	218
10.4	Модуль sys — интерфейс к системе исполнения	219
10.4.1	Запрос информации о текущей системной среде	220
10.4.2	Получение аргументов командной строки	221
10.4.3	За кулисами: управление памятью	223
10.4.4	Доступ к модулям	225
10.4.5	Перенаправление стандартного вывода в файл	226
10.5	Повторение	227
10.6	Упражнения	227
10.7	Ответ на вопрос. Какой комментарий верный?	229
Глава 11	Графические пользовательские интерфейсы	230
11.1	Виджеты	231
11.2	Окно приложения Tk	231
11.3	Вставка виджета	233
11.4	Дизайн внешнего вида виджетов	233
11.4.1	Размер виджета	235
11.5	Общие методы виджетов	236
11.6	Кнопки и обработчики событий	236

11.6.1	Проект «Мотиватор»	237
11.7	Изменение макета	238
11.8	Растровый макет	241
11.9	Проект «25 цветов — изображение с автоматическими образцами»	242
11.10	Виджеты ввода текста	244
11.10.1	Однорочный ввод: виджет «Ввод»	244
11.10.2	Многострочный ввод: текстовый виджет	246
11.10.3	Проект «Рифмы с Гете»	248
11.11	Радиокнопки	249
11.11.1	Проект «Конвертер валюты»	250
11.12	Диалоговые окна	252
11.12.1	Проект «Текстовый редактор»	253
11.13	Параллельные процессы: потоки (threads)	254
11.13.1	Эксперимент «Обратный отсчет»	255
11.13.2	Запуск функции в собственном потоке	256
11.14	Повторение	257
11.15	Упражнения	258
11.16	Ответы на вопросы	260
Глава 12	Программирование графиков	262
12.1	Изображения на кнопках и метках	262
12.1.1	Проект «Игра в кости»	262
12.1.2	Редактирование изображения	265
12.1.3	Проект «Оттенки серого»	265
12.2	Холст	267
12.2.1	Создание графических элементов	268
12.2.2	Создание линий	270
12.2.3	Идентификационные номера: удаление или перемещение элементов	271
12.3	Проект «Креативное кодирование»	271
12.4	Библиотека изображений Python (PIL, Python Imaging Library) ...	274
12.4.1	Получение объекта изображения из файла	274
12.4.2	Создание объекта изображения без файла	276
12.4.3	Атрибуты и методы объектов изображения	276
12.4.4	Обработка изображений с помощью списков	278
12.4.5	Вставка изображений	280
12.4.6	Проект «Зеленый экран»	280
12.4.7	Объекты PIL.Image в приложениях tkinter	283
12.4.8	Проект «Просмотр веб-камеры»	283
12.5	Повторение	285
12.6	Упражнения	285

Глава 13 Поиск ошибок и возможности их не допускать	289
13.1 Утверждения	289
13.2 Трассировка	291
13.2.1 Пример: быстрая сортировка (quicksort)	292
13.3 Отладка с помощью IDLE	294
13.3.1 Отладчик оболочки Python	294
13.3.2 Пошаговое выполнение программы	295
13.3.3 Установка точек останова	296
13.4 Отладка с Thonny	297
13.5 Повторение	299
13.6 Ответы на вопросы	300
Глава 14 Объектно-ориентированное программирование	301
14.1 Классы и объекты	301
14.1.1 Что такое объектная ориентация?	301
14.1.2 Проектирование и графическое отображение классов — UML	302
14.1.3 Определение класса	303
14.1.4 Создание объектов класса: инстанцирование	305
14.1.5 Атрибуты	305
14.1.6 Вызов методов	305
14.1.7 Объекты с переменными начальными значениями	306
14.1.8 Метафоры в программировании	306
14.2 Проект «Деньги»	307
14.2.1 Расчеты с денежными объектами	308
14.2.2 Атрибуты класса	309
14.2.3 Перегрузка операторов — полиморфизм	309
14.3 «Магические» методы	312
14.4 Проект «Расчет»	313
14.5 Наследование	315
14.6 Проект «Тестер цвета»	318
14.7 Проект «Числовой дождь»	321
14.8 Повторение	325
14.9 Упражнения	325
14.10 Ответы на вопросы	328
Глава 15 Базы данных	330
15.1 Что такое система базы данных?	330
15.2 Проектирование базы данных — диаграмма Entity-Relationship (ER)	330
15.3 Реляционные базы данных	332
15.4 Представление связей в Python*	334

15.4.1	Набор кортежей	334
15.4.2	Именованные кортежи	334
15.5	Модуль <code>sqlite3</code> — интерфейс к базе данных SQL	336
15.5.1	Создание таблиц с помощью SQL и ввод кортежей	336
15.5.2	Создание базы данных SQLite с помощью <code>sqlite3</code>	337
15.5.3	Формулировка запросов с помощью SQL	339
15.5.4	Запросы к базе данных в программах Python	340
15.5.5	Операторы SQL с переменными частями	342
15.6	Проект «Сборник цитат»	342
15.6.1	ER-диаграмма	343
15.6.2	Пример: таблицы	343
15.6.3	Работа со сборником цитат	344
15.6.4	Поиск цитат	347
15.7	Повторение	350
15.8	Упражнения	351
15.9	Ответы на вопросы	352
Глава 16	Научные проекты	354
16.1	NumPy — вычисления с массивами	354
16.1.1	Массивы	354
16.1.2	Индексирование	359
16.1.3	Нарезка	361
16.1.4	Изменение массивов	362
16.1.5	Арифметические операции	364
16.1.6	Функции, выполняемые поэлементно	365
16.1.7	Умножение матриц с помощью <code>dot()</code>	365
16.1.8	Функции массива и оси	366
16.2	Визуализация данных с помощью <code>matplotlib</code>	368
16.2.1	Линейные графики	370
16.2.2	Несколько линий на диаграмме	371
16.2.3	Гистограммы	372
16.2.4	Проект «Игра в кости»	374
16.2.5	Тепловые карты (heatmaps)	375
16.3	Проект «Профиль движения»	376
6.4	Google Collaborative — Colab	380
16.4.1	Создание блокнота Colab	380
16.4.2	Текстовые ячейки	383
16.4.3	Вставка изображений	384
16.4.4	Сохранение и открытие блокнотов	385
16.5	Проект «Лисы и зайцы — моделирование системы „хищник-жертва“»	386
16.5.1	Совместное использование блокнотов	389

16.6	Повторение	390
16.7	Упражнения	391
16.8	Ответы на вопросы	393
Глава 17	Динамические веб-сайты: CGI и WSGI	396
17.1	Динамические веб-страницы с использованием CGI	398
17.1.1	Проект «Который час?»	399
17.1.2	Вывод сценария CGI	399
17.1.3	Как устроен сценарий CGI?	400
17.1.4	CGI-скрипты в Windows	401
17.1.5	Вызов с помощью веб-браузера	401
17.1.6	HTTP-сервер	402
17.1.7	Доступ с другого компьютера в локальной сети	403
17.2	Интерактивные веб-сайты	404
17.2.1	Компоненты ввода в форме HTML	405
17.2.2	Обработка входных данных с помощью FieldStorage	407
17.3	Как обрабатывать умлауты? *	409
17.4	Динамические веб-страницы с WSGI	410
17.4.1	Объект приложения	411
17.4.2	Скрипты с собственным HTTP-сервером — модуль wsgiref	411
17.5	Проект «Который час? (2)»	412
17.6	Проект «Опрос»	414
17.6.1	Шаблоны HTML	416
17.6.2	Алгоритмическая часть	417
17.7	Использование хостинга	419
17.7.1	Python Anywhere	419
17.7.2	Опробование готовой программы WSGI	419
17.7.3	Проект «Который час? (3)»	422
17.7.4	Модуляризация проектов WSGI	423
17.8	Повторение	424
17.9	Упражнения	425
17.10	Ответ на вопрос. Интерактивный сайт	427
Глава 18	Профессиональная разработка программного обеспечения	428
18.1	Срок действия программ	428
18.1.1	Быстрая сортировка — Quicksort vs. Straight Selection	428
18.1.2	Анализ производительности с помощью профилировщика	431
18.2	Гибкая разработка программного обеспечения	433
18.2.1	Разработка программного обеспечения	433

18.2.2	Некоторые основные идеи гибкой разработки программного обеспечения	434
18.3	Проект «Цифровой блокнот»	435
18.3.1	Stories	436
18.3.2	Первая итерация	436
18.3.3	Вторая итерация	437
18.3.4	Рефакторинг	439
18.3.5	Новые истории и возможность вносить изменения	442
18.4	Разработка через тестирование (Test Driven Development) с помощью doctest	444
18.5	Упражнение «Бронирование билетов»	446
	Глоссарий	448
	Предметный указатель	457

Введение

Во многих профессиях, в том числе не связанных с информационными технологиями, знание программирования — необходимый базовый навык. Даже если вы не занимаетесь профессиональной разработкой программного обеспечения (ПО), вам может понадобиться написать компьютерную программу или поучаствовать в разработках в рамках научных проектов или в своей профессиональной практике. Кроме того, программирование отлично формирует логическое мышление. Те, кто в этом подкован, умеют лучше анализировать проблемы и находить решения.

Это пособие предназначено в первую очередь для тех, кому для учебы или в рамках профессиональной подготовки необходимо познакомиться с программированием на Python. Пособие можно использовать как в качестве основного материала для курса программирования, так и для самостоятельного обучения.

На примере приведенных в книге объяснений вы легко разберетесь в этой теме — они не требуют каких-либо предварительных знаний. Удобнее всего читать недалеко от компьютера, чтобы иметь возможность сразу же опробовать примеры программ. Многочисленные практические упражнения по программированию помогут вам усвоить новые знания. Вы быстро добьетесь первых успехов и получите удовольствие от программирования.

Структура

Пособие начинается с основ: установки Python, использования среды разработки и написания простых команд. Вы шаг за шагом научитесь загружать, обрабатывать и сохранять данные, а также познакомитесь с использованием функций и модулей, объектно-ориентированным программированием и разработкой графических пользовательских интерфейсов.

В последующих главах вы сможете применить изученные концепции Python в более сложных и интересных областях компьютерных наук: в технологии баз данных, обработке изображений, научных вычислениях с помощью NumPy, визуализации данных с помощью Matplotlib и в интернет-программировании.

Вы можете пропускать разделы, отмеченные звездочкой *, если тема вам не интересна. Они посвящены вопросам, проработка которых не требуется для понимания следующей темы.

Последняя глава дает представление о передовых методах (например, поиск уязвимостей в программе с помощью анализа производительности) и некоторых идеях гибких методик разработки, которые могут помочь успешно спланировать и реализовать более крупный программный проект.

Иногда на страницах пособия вам будут встречаться дополнительные вопросы. Это своего рода обучающая деятельность, и ответы на эти вопросы будут даны в конце главы. Каждая глава завершается практическими упражнениями по программированию, с помощью которых вы сможете углубить новые знания. Уровень сложности заданий отмечен звездочкой *. Чем больше звездочек, тем сложнее задание. Решения упражнений, которые содержат много текста программы, доступны для скачивания в *онлайн-главе*. Подробнее об этом в следующем разделе.

В конце пособия вы найдете глоссарий с наиболее важными техническими терминами и предметный указатель, который поможет быстрее находить конкретные темы.

Обратите внимание на шрифт!

В нашем пособии шрифт также имеет значение. Так вам будет легче с ним взаимодействовать. Все тексты программ или их части (например, имена переменных) набираются моноширинным шрифтом.

Пример: переменная `name` имеет значение `'Jessy'`.

В некоторых абзацах программных текстов появляются выделенные *курсивом* моноширинные тексты, которые означают подстановочный символ. В программе вы должны заменить его другим текстом, который соответствует контексту.

Пример:

```
stream = open(имя файла)
```

И здесь *stream* и *имя файла* будут как раз такими подстановочными символами.

Если речь идет о диалоге с компьютером, то текст, который введен человеком, выглядит немного светлее, чем текст, который выводит компьютер.