

Оглавление

От издательства	25
Введение.....	26
Предисловие	28
Благодарности.....	30
Об этой книге	32
Об авторе.....	37
Об иллюстрации на обложке	38

Часть 1 Первые шаги

Глава 1. Знакомство с Python.....	40
1.1. Почему стоит использовать Python?	41
1.2. С чем хорошо справляется Python.....	41
1.2.1. Python прост в использовании	42
1.2.2. Выразительность Python.....	42
1.2.3. Удобочитаемость кода Python	43
1.2.4. Полнота Python: «батарейки в комплекте»	44
1.2.5. Богатая экосистема сторонних библиотек	44
1.2.6. Кросс-платформенность	44
1.2.7. Свободное распространение.....	45
1.3. Что в Python стоит улучшить	46
1.3.1. Не самый быстрый язык.....	46

1.3.2. Python не применяет принудительное определение типов переменных во время компиляции.....	46
1.3.3. Слабая поддержка мобильных устройств	47
1.3.4. Слабая поддержка многоядерных систем	48
Итоги.....	48
Глава 2. Первые шаги	49
2.1. Парадокс выбора: вам какой Python?	49
2.1.1. Версии Python.....	50
2.1.2. Как скачать Python	50
2.1.3. Устройства, платформы и операционные системы	51
2.1.4. Среды и инструменты.....	51
2.2. Colaboratory: Jupyter Notebooks в облаке	52
2.2.1. Доступ к исходному коду блокнотов	52
2.2.2. Начало работы с Colaboratory	52
2.2.3. Версия Python для Colaboratory.....	55
2.3. Написание и запуск программ в Colaboratory	56
2.3.1. Hello, World	56
2.3.2. Работа с ошибками в Jupyter.....	57
2.4. Применение функций help и dir для изучения Python.....	58
2.5. Использование инструментов ИИ для написания кода на Python.....	61
2.5.1. Преимущества инструментов ИИ.....	61
2.5.2. Недостатки инструментов ИИ	61
2.5.3. Варианты ИИ-инструментов	62
Итоги.....	63
Глава 3. Краткий обзор Python	64
3.1. Краткая характеристика Python	64
3.2. Встроенные типы данных.....	65
3.2.1. Числа	65
3.2.2. Списки	68
3.2.3. Кортежи.....	70
3.2.4. Строки.....	70
3.2.5. Словари.....	72

3.2.6. Множества	72
3.2.7. Файловые объекты	73
3.3. Аннотации типов в Python	74
3.4. Управляющие конструкции.....	74
3.4.1. Логические значения и выражения	74
3.4.2. Оператор if-elif-else.....	75
3.4.3. Структурное сопоставление шаблонов при помощи оператора match.....	75
3.4.4. Цикл while.....	76
3.4.5. Цикл for	76
3.4.6. Определение функции	77
3.4.7. Исключения	78
3.4.8. Обработка контекста с использованием ключевого слова with	79
3.5. Создание модуля	79
3.6. Объектно-ориентированное программирование	81
Итоги.....	82

Часть 2

Основы

Глава 4. Основы основ	84
4.1. Отступы и блочная структура	84
4.2. Как определить комментарий.....	86
4.3. Переменные и присваивания.....	86
4.4. Необязательные аннотации типов в Python.....	89
4.4.1. Когда использовать аннотации типов?	91
4.4.2. Когда не использовать аннотации типов?	91
4.4.3. Прогрессивная типизация.....	91
4.5. Выражения	92
4.6. Строки.....	92
4.7. Числа	93
4.7.1. Встроенные числовые функции.....	95
4.7.2. Расширенные числовые функции	95
4.7.3. Числовые вычисления.....	96

4.7.4. Комплексные числа.....	96
4.7.5. Расширенные функции комплексных чисел.....	97
4.8. Значение None.....	98
4.9. Получение пользовательского ввода	98
4.10. Встроенные операторы	99
4.11. Основные элементы стиля Python	99
Итоги.....	100
Глава 5. Списки, кортежи и множества	101
5.1. Списки подобны массивам	102
5.2. Индексы списка	103
5.3. Модификация списков.....	105
5.4. Сортировка списков.....	108
5.4.1. Пользовательская сортировка.....	109
5.4.2. Функция sorted()	111
5.5. Другие операции со списками	111
5.5.1. Проверка принадлежности элемента списку оператором in	111
5.5.2. Конкатенация списков оператором +	112
5.5.3. Инициализация списков оператором *	112
5.5.4. Получение наименьшего или наибольшего элемента функциями min и max	113
5.5.5. Поиск по списку при помощи метода index.....	113
5.5.6. Подсчет вхождений методом count.....	114
5.5.7. Краткое описание операций со списками.....	114
5.6. Вложенные списки и глубокие копии	115
5.7. Кортежи.....	117
5.7.1. Базовые сведения о кортежах.....	118
5.7.2. Одноэлементные кортежи должны содержать запятую.....	120
5.7.3. Упаковка и распаковка кортежей	120
5.7.4. Преобразования между списками и кортежами.....	122
5.8. Множества.....	123
5.8.1. Операции с множествами.....	123
5.8.2. Замороженные множества	124

5.9. Практическая работа: изучение списка.....	125
5.9.1. Зачем решать эту задачу устаревшим способом?	125
5.9.2. Решение задачи при помощи кода, сгенерированного ИИ	126
5.9.3. Решения и обсуждение.....	127
Итоги.....	129
Глава 6. Строки.....	130
6.1. Строки как последовательности символов.....	130
6.2. Основные операции со строками.....	131
6.3. Специальные символы и экранирующие последовательности	132
6.3.1. Основные экранирующие последовательности	132
6.3.2. Числовые (восьмеричные и шестнадцатеричные) и экранирующие последовательности Unicode.....	133
6.3.3. Вывод и обработка строк со специальными символами	134
6.4. Строковые методы.....	135
6.4.1. Строковые методы split и join.....	135
6.4.2. Преобразование строк в числа.....	137
6.4.3. Удаление лишних пробельных символов	138
6.4.4. Поиск в строках	140
6.4.5. Изменение строк	142
6.4.6. Изменение строк при помощи операций со списками.....	144
6.4.7. Полезные методы и константы.....	145
6.5. Преобразование объектов в строки	146
6.6. Использование метода format.....	148
6.6.1. Метод format и позиционные параметры	148
6.6.2. Метод format и именованные параметры	149
6.6.3. Спецификаторы формата.....	149
6.7. Интерполяция строк с помощью f-строк	150
6.8. Форматирование строк с помощью %.....	151
6.8.1. Применение форматирующих последовательностей.....	152
6.8.2. Именованные параметры и форматирующие последовательности	153
6.9. Байтовые объекты.....	154

6.10. Практическая работа: предварительная обработка текста.....	155
6.10.1. Решение задачи при помощи кода, сгенерированного ИИ.....	156
6.10.2. Решения и обсуждение	157
Итоги	159
Глава 7. Словари	160
7.1. Что такое словарь?.....	161
7.2. Другие операции со словарями	163
7.3. Подсчет слов	166
7.4. Что может служить ключом?	167
7.5. Разреженные матрицы	168
7.6. Словари как кэши	170
7.7. Эффективность словарей	171
7.8. Практическая работа: подсчет слов	171
7.8.1. Решение задачи с помощью кода, сгенерированного ИИ.....	171
7.8.2. Решения и обсуждение.....	172
Итоги.....	175
Глава 8. Управляющие конструкции	176
8.1. Оператор if-elif-else.....	176
8.2. Структурное сопоставление шаблонов с помощью оператора match.....	178
8.3. Цикл while.....	179
8.4. Цикл for	180
8.4.1. Функция range	181
8.5. Управление диапазоном с помощью аргументов для начала и шага... 182	
8.6. Цикл for и распаковка кортежа	183
8.7. Функция enumerate	183
8.8. Функция zip	183
8.9. Генераторы списков, множеств и словарей	184
8.9.1. Генераторные выражения.....	185
8.10. Операторы, блоки и отступы	186

8.11. Логические значения и выражения	189
8.11.1. Использование объектов Python как логических значений	190
8.11.2. Операторы сравнения и логические операторы	190
8.12. Простая программа для анализа текстового файла	192
8.13. Практическая работа: рефакторинг word_count.....	193
8.13.1. Решение задачи при помощи кода, сгенерированного ИИ	193
8.13.2. Решения и обсуждение	194
Итоги.....	196
Глава 9. Функции	197
9.1. Базовые определения функций.....	197
9.2. Параметры функций	199
9.2.1. Позиционные параметры	199
9.2.2. Передача аргументов по имени параметра.....	200
9.2.3. Переменное количество аргументов	201
9.2.4. Совмещение способов передачи аргументов.....	203
9.3. Изменяемые объекты в качестве аргументов.....	203
9.3.1. Изменяемые объекты в качестве значений по умолчанию	204
9.4. Локальные, нелокальные и глобальные переменные	206
9.5. Присваивание функций переменным	208
9.6. Лямбда-выражения	209
9.7. Функции-генераторы	210
9.8. Декораторы.....	211
9.9. Практическая работа: полезные функции.....	213
9.9.1. Решение задачи при помощи кода, сгенерированного ИИ	213
9.9.2. Решения и обсуждение.....	213
Итоги	220
Глава 10. Модули и правила области видимости	221
10.1. Что такое модуль?	221
10.2. Первый модуль	222
10.3. Оператор import	226

10.4. Путь поиска модулей.....	227
10.4.1. Где следует размещать собственные модули.....	228
10.5. Приватные имена в модулях.....	230
10.6. Модули стандартной библиотеки Python и сторонние.....	231
10.7. Правила области видимости и пространств имен в языке Python....	232
10.7.1. Встроенное пространство имен	236
10.8. Практическая работа: создание модуля	239
10.8.1. Решение задачи при помощи кода, сгенерированного ИИ.....	240
10.8.2. Решения и обсуждение	240
Итоги.....	248
Глава 11. Программы на Python.....	249
11.1. Создание простейшей программы	250
11.1.1. Запуск скрипта из командной строки	250
11.1.2. Аргументы командной строки.....	251
11.1.3. Выполнение кода только как основного скрипта.....	252
11.1.4. Перенаправление ввода и вывода скрипта	253
11.1.5. Модуль argparse.....	254
11.1.6. Применение модуля fileinput.....	256
11.2. Запуск скриптов в разных операционных системах.....	258
11.2.1. Создание скрипта, исполняемого непосредственно в UNIX....	258
11.2.2. Скрипты на macOS.....	259
11.2.3. Параметры выполнения скриптов в Windows.....	259
11.3. Программы и модули.....	260
11.4. Распространение приложений на Python	266
11.4.1. Wheel-пакеты	266
11.4.2. zipapp и rex.....	266
11.4.3. py2exe и py2app.....	267
11.4.4. Создание исполняемых программ с помощью freeze	267
11.5. Практическая работа: создание программы	268
11.5.1. Решение задачи при помощи кода, сгенерированного ИИ.....	268
11.5.2. Решения и обсуждение	269
Итоги	276

Глава 12. Работа с файловой системой	277
12.1. Модули <code>os</code> и <code>os.path</code> в сравнении с <code>pathlib</code>	278
12.2. Пути и путевые имена	278
12.2.1. Абсолютные и относительные пути	279
12.2.2. Текущий рабочий каталог	280
12.2.3. Обращение к каталогам с помощью <code>pathlib</code>	281
12.2.4. Управление путями	282
12.2.5. Управление путями с помощью <code>pathlib</code>	284
12.2.6. Полезные константы и функции	286
12.3. Получение информации о файлах	287
12.3.1. Получение информации о файлах с помощью <code>scandir</code>	288
12.4. Другие операции с файловой системой	289
12.4.1. Другие операции с файловой системой при помощи <code>pathlib</code> ...	292
12.5. Обработка всех файлов в поддереве каталога	294
12.6. Практическая работа: другие операции с файлами	295
12.6.1. Решение задачи при помощи кода, сгенерированного ИИ	296
12.6.2. Решения и обсуждение	296
Итоги	304
Глава 13. Чтение и запись файлов	307
13.1. Открытие файлов и файловые объекты	308
13.2. Закрывтие файлов	309
13.3. Открытие файлов для записи и в других режимах	309
13.4. Функции для чтения и записи текстовых и двоичных данных	310
13.4.1. Двоичный режим	312
13.5. Чтение и запись при помощи <code>pathlib</code>	313
13.6. Консольный ввод-вывод и перенаправление	314
13.7. Обработка структурированных двоичных данных с помощью модуля <code>struct</code>	317
13.8. Пиклинг объектов в файлы	320
13.8.1. Когда лучше обойтись без пиклинга	323
13.9. Сохранение объектов в модуле <code>shelve</code>	324
13.10. Практическая работа: последние доработки утилиты <code>wc</code>	326

13.10.1. Решение задачи при помощи кода, сгенерированного ИИ....	326
13.10.2. Решения и обсуждение	327
Итоги.....	333
Глава 14. Исключения.....	334
14.1. Введение в исключения.....	335
14.1.1. Общие принципы обработки ошибок и исключений.....	335
14.1.2. Более формальное определение исключений.....	337
14.1.3. Обработка разных типов исключений.....	338
14.2. Исключения в Python	338
14.2.1. Типы исключений в языке Python.....	339
14.2.2. Возникновение исключений.....	341
14.2.3. Перехват и обработка исключений.....	342
14.2.4. Установление новых исключений.....	344
14.2.5. Группы исключений.....	345
14.2.6. Отладка программ с помощью оператора assert	346
14.2.7. Иерархия наследования исключений.....	347
14.2.8. Пример: программа Python для записи на диск.....	347
14.2.9. Пример: исключения в обычных вычислениях.....	348
14.2.10. Где можно использовать исключения	350
14.3. Контекстные менеджеры с ключевым словом with	350
14.4. Практическая работа: добавление исключений	352
14.4.1. Решение задачи при помощи кода, сгенерированного ИИ.....	352
14.4.2. Решения и обсуждение	352
Итоги	360

Часть 3

Расширенные возможности языка

Глава 15. Классы и объектно-ориентированное программирование.....	362
15.1. Определение классов.....	362
15.1.1. Использование экземпляра класса как структуры или записи.....	363
15.2. Переменные экземпляра	364
15.3. Методы	365

15.4. Классовые переменные	367
15.4.1. Необычное поведение классовых переменных	369
15.5. Статические методы и методы классов.....	370
15.5.1. Статические методы	370
15.5.2. Методы класса	371
15.6. Наследование	372
15.7. Наследование с использованием классовых переменных и переменных экземпляра	375
15.8. Резюмируем: основные возможности классов Python	376
15.9. Закрытые переменные и закрытые методы	379
15.10. Использование декоратора @property для гибкой работы с переменными экземпляра.....	380
15.11. Правила определения области видимости и пространств имен для экземпляров класса.....	382
15.12. Деструкторы и управление памятью	386
15.13. Множественное наследование	386
15.14. Практическая работа: HTML-классы.....	389
15.14.1. Решение задачи при помощи кода, сгенерированного ИИ.....	389
15.14.2. Решения и обсуждение	390
Итоги.....	393
Глава 16. Регулярные выражения.....	394
16.1. Что такое регулярное выражение?	394
16.2. Регулярные выражения со специальными символами	395
16.3. Регулярные выражения и сырые (необработанные) строки	397
16.3.1. Сырые строки спешат на помощь	398
16.4. Извлечение из строк текста, совпадающего с шаблоном	399
16.5. Замена текста при помощи регулярных выражений.....	403
16.5.1. Использование функции с методом sub	403
16.6. Практическая работа: нормализация телефонных номеров.....	404
16.6.1. Решение задачи при помощи кода, сгенерированного ИИ	405
16.6.2. Решения и обсуждение	405
Итоги.....	410

Глава 17. Типы данных как объекты	412
17.1. Типы данных тоже являются объектами	412
17.2. Работа с типами данных	413
17.3. Типы данных и пользовательские классы.....	414
17.4. Утиная (неявная) типизация	416
17.5. Что такое специальный метод?	417
17.6. Придание объекту свойств списка.....	418
17.7. Специальный метод <code>__getitem__</code>	419
17.7.1. Как это работает	420
17.7.2. Реализация полной функциональности списка.....	421
17.8. Наделение объекта полной функциональностью списка	422
17.9. Наследование от встроенных типов.....	425
17.9.1. Наследование от <code>list</code>	425
17.9.2. Наследование от <code>UserList</code>	426
17.10. Когда следует применять специальные методы	427
17.11. Практическая работа: создание словаря со строковыми ключами.....	428
17.11.1. Решение задачи при помощи кода, сгенерированного ИИ.....	429
17.11.2. Решения и обсуждение	429
Итоги.....	432
Глава 18. Пакеты	433
18.1. Что такое пакет?	434
18.2. Первый пример: <code>mathproj</code>	434
18.3. Реализация пакета <code>mathproj</code>	435
18.3.1. Файлы <code>__init__.py</code> в пакетах	437
18.3.2. Основы использования пакета <code>mathproj</code>	438
18.3.3. Загрузка подпакетов и подмодулей.....	438
18.3.4. Операторы <code>import</code> внутри пакетов.....	439
18.4. Атрибут <code>__all__</code>	440
18.5. Правильное использование пакетов	442
18.6. Практическая работа: создание пакета.....	442
18.6.1. Решение задачи при помощи кода, сгенерированного ИИ	442

18.6.2. Решения и обсуждение	443
Итоги.....	455
Глава 19. Работа с библиотеками Python	456
19.1. «Все включено»: стандартная библиотека	457
19.1.1. Управление различными типами данных.....	457
19.1.2. Файловые операции и хранение данных.....	459
19.1.3. Доступ к службам операционной системы	459
19.1.4. Работа с интернет-протоколами и форматами данных	460
19.1.5. Средства разработки и отладки, службы времени выполнения	461
19.2. За пределами стандартной библиотеки	462
19.3. Установка дополнительных библиотек Python.....	462
19.4. Индекс пакетов Python (The Python Package Index)	463
19.5. Установка библиотек Python при помощи pip и venv	463
19.5.1. Установка с флагом --user	464
19.5.2. Виртуальные среды.....	465
19.5.3. Прочие варианты	466
Итоги	466

Часть 4

Работа с данными

Глава 20. Общая подготовка файлов	468
20.1. Проблема: бесконечный поток файлов данных	468
20.2. Сценарий: кошмарный товарный фид.....	469
20.3. Дальнейшее структурирование.....	472
20.4. Оптимизация хранения: сжатие и упорядочивание	473
20.4.1. Сжатие файлов	473
20.4.2. Упорядочивание файлов	475
Итоги.....	477
Глава 21. Обработка файлов данных	478
21.1. Знакомство с ETL.....	479

21.2. Чтение текстовых файлов.....	479
21.2.1. Текстовые кодировки: ASCII, Unicode и другие.....	479
21.2.2. Неструктурированный текст	481
21.2.3. Плоские (простые текстовые) файлы с разделителями.....	484
21.2.4. Модуль csv.....	486
21.2.5. Чтение csv-файла как списка словарей.....	489
21.3. Файлы Excel	489
21.4. Очистка данных.....	492
21.4.1. Очистка	492
21.4.2. Сортировка.....	493
21.4.3. Сложности и подводные камни очистки данных	494
21.5. Запись файлов данных.....	495
21.5.1. CSV и другие файлы с разделителями.....	495
21.5.2. Запись файлов Excel.....	496
21.5.3. Архивирование файлов данных.....	497
21.6. Практическая работа: метеорологические наблюдения.....	497
21.6.1. Решение задачи при помощи кода, сгенерированного ИИ.....	498
21.6.2. Решения и обсуждение	498
Итоги.....	502
Глава 22. Получение данных по сети	503
22.1. Загрузка файлов.....	503
22.1.1. Получение файлов с FTP-сервера с помощью Python	504
22.1.2. Получение файлов с помощью SFTP	506
22.1.3. Получение файлов по протоколу HTTP/HTTPS.....	506
22.2. Получение данных через API.....	508
22.3. Структурированные форматы данных.....	510
22.3.1. Данные в формате JSON	511
22.3.2. Данные в формате XML	515
22.4. Сбор данных с веб-страниц (скрейпинг).....	518
22.5. Практическая работа: сбор метеоданных	522
22.5.1. Решение задачи при помощи кода, сгенерированного ИИ.....	523
22.5.2. Решения и обсуждение	523
Итоги.....	526

Глава 23. Сохранение данных.....	527
23.1. Реляционные базы данных.....	528
23.1.1. API баз данных (DB-API) в Python	528
23.2. SQLite: работа с базой данных sqlite3.....	528
23.3. MySQL, PostgreSQL и другие реляционные базы данных	531
23.4. Оптимизация работы с базами данных при помощи объектно- реляционного отображения (ORM)	532
23.4.1. SQLAlchemy.....	532
23.4.2. Изменение схемы базы данных при помощи Alembic	537
23.5. Нереляционные базы данных (NoSQL).....	540
23.6. Хранилища «ключ–значение» на примере Redis	541
23.7. Документы в MongoDB.....	545
23.8. Практическая работа: создание базы данных	550
Итоги.....	550
Глава 24. Анализ данных.....	551
24.1. Средства Python для анализа данных	551
24.1.1. Преимущества Python для анализа данных	551
24.1.2. Python может быть эффективнее электронных таблиц	552
24.2. Python и pandas.....	552
24.2.1. Почему стоит использовать pandas?	553
24.2.2. Установка pandas	553
24.2.3. Датафреймы.....	554
24.3. Очистка данных.....	556
24.3.1. Загрузка и сохранение данных с помощью pandas.....	556
24.3.2. Очистка данных при помощи датафрейма.....	558
24.4. Агрегирование и обработка данных	561
24.4.1. Слияние датафреймов.....	561
24.4.2. Выборка данных	562
24.4.3. Группировка и агрегирование.....	563
24.5. Построение графика данных.....	565
24.6. Когда не стоит использовать pandas	566
Итоги.....	567

Практический пример	568
Загрузка данных.....	569
Парсинг данных по периодам наблюдений (инвентаризациям).....	572
Выбор станции по широте и долготе.....	574
Выбор станции и получение ее метаданных.....	576
Получение и парсинг метеоданных.....	578
Получение данных.....	578
Парсинг метеоданных.....	579
Сохранение метеоданных в базе данных (необязательно).....	582
Выборка данных и построение графиков	583
Использование pandas для графического представления данных.....	584
Приложение. Руководство по документации Python	585
П.1. Доступ к онлайн-документации Python	585
П.2. Лучшие практики: как стать питонистом.....	586
П.2.1. Десять советов о том, как стать питонистом	586
П.3. PEP 8: Стандарт стилевого оформления кода Python.....	588
П.3.1. Введение.....	588
П.3.2. Оформление кода.....	589
П.4. Комментарии.....	594
П.4.1. Правила именования	596
П.4.2. Рекомендации по программированию.....	602
П.4.3. Другие руководства по стилю Python	605
П.5. Дзен Python.....	605