
Краткое содержание

Введение.....	26
Благодарности.....	33
Об авторе.....	34

Часть I. Основы Python

Глава 1. Python: с чем его едят	36
Глава 2. Данные: типы, значения, переменные и имена.....	55
Глава 3. Числа	67
Глава 4. Выбираем с помощью оператора if	82
Глава 5. Текстовые строки	91
Глава 6. Создаем циклы с помощью ключевых слов while и for.....	113
Глава 7. Кортежи и списки	119
Глава 8. Словари и множества	144
Глава 9. Функции	166
Глава 10. Ой-ой-ой: объекты и классы	194
Глава 11. Модули, пакеты и программы	224

Часть II. Python на практике

Глава 12. Обработываем данные.....	242
Глава 13. Календари и часы.....	271
Глава 14. Файлы и каталоги.....	283
Глава 15. Данные во времени: процессы и конкурентность	302

Глава 16. Данные в коробке: надежные хранилища	327
Глава 17. Данные в пространстве: сети	368
Глава 18. Распутываем Всемирную паутину	400
Глава 19. Быть питонщиком	431
Глава 20. Пи-Арт	474
Глава 21. За работой	487
Глава 22. Python в науке.....	503

Приложения

Приложение А. Аппаратное и программное обеспечение для начинающих программистов	520
Приложение Б. Установка Python 3	530
Приложение В. Нечто совершенно иное: async.....	538
Приложение Г. Ответы к упражнениям	544
Приложение Д. Вспомогательные таблицы	587
Эпилог	591

Оглавление

Введение.....	26
Для кого эта книга.....	27
Что нового во втором издании.....	27
Структура книги.....	28
Версии Python.....	31
Условные обозначения.....	31
Использование примеров кода.....	32
От издательства.....	32
Благодарности.....	33
Об авторе.....	34

Часть I. Основы Python

Глава 1. Python: с чем его едят	36
Тайны.....	36
Маленькие программы.....	38
Более объемная программа.....	40
Python в реальном мире.....	44
Python против языка с планеты X.....	45
Почему же Python?.....	48
Когда не стоит использовать Python.....	49
Python 2 против Python 3.....	50
Установка Python.....	51
Запуск Python.....	51
Интерактивный интерпретатор.....	51
Файлы Python.....	52
Что дальше?.....	53

Момент просветления.....	53
Читайте далее.....	54
Упражнения.....	54
Глава 2. Данные: типы, значения, переменные и имена	55
В Python данные являются объектами	55
Типы	56
Изменчивость	57
Значения-литералы	58
Переменные.....	58
Присваивание	60
Переменные — это имена, а не локации.....	61
Присваивание нескольким именам.....	64
Переназначение имени	64
Копирование	64
Выбираем хорошее имя переменной	65
Читайте далее.....	66
Упражнения.....	66
Глава 3. Числа	67
Булевы значения.....	67
Целые числа.....	68
Числа-литералы.....	68
Операции с целыми числами.....	69
Целые числа и переменные	71
Приоритет операций.....	73
Системы счисления.....	74
Преобразования типов	76
Насколько объемён тип int.....	78
Числа с плавающей точкой	79
Математические функции.....	80
Читайте далее.....	81
Упражнения.....	81
Глава 4. Выбираем с помощью оператора if	82
Комментируем с помощью символа #.....	82
Продлеваем строки с помощью символа \.....	83
Сравниваем с помощью операторов if, elif и else	84
Что есть истина?	87
Выполняем несколько сравнений с помощью оператора in	88

Новое: I Am the Walrus	89
Читайте далее.....	90
Упражнения	90
Глава 5. Текстовые строки	91
Создаем строки с помощью кавычек	91
Создаем строки с помощью функции <code>str()</code>	94
Создаем escape-последовательности с помощью символа <code>\</code>	94
Объединяем строки с использованием символа <code>+</code>	96
Размножаем строки с помощью символа <code>*</code>	96
Извлекаем символ с помощью символов <code>[]</code>	97
Извлекаем подстроки, используя разделение	98
Измеряем длину строки с помощью функции <code>len()</code>	100
Разделяем строку с помощью функции <code>split()</code>	100
Объединяем строки с помощью функции <code>join()</code>	101
Заменяем символы с использованием функции <code>replace()</code>	101
Устраняем символы с помощью функции <code>strip()</code>	102
Поиск и выбор	103
Регистр	104
Выравнивание.....	105
Форматирование.....	105
Старый стиль: <code>%</code>	106
Новый стиль: используем символы <code>{ }</code> и функцию <code>format()</code>	108
Самый новый стиль: f-строки	110
Что еще можно делать со строками.....	111
Читайте далее.....	111
Упражнения	111
Глава 6. Создаем циклы с помощью ключевых слов <code>while</code> и <code>for</code>.....	113
Повторяем действия с помощью цикла <code>while</code>	113
Прерываем цикл с помощью оператора <code>break</code>	114
Пропускаем итерации, используя оператор <code>continue</code>	114
Проверяем, завершился ли цикл раньше, с помощью блока <code>else</code>	115
Выполняем итерации с использованием ключевых слов <code>for</code> и <code>in</code>	115
Прерываем цикл с помощью оператора <code>break</code>	116
Пропускаем итерации, используя оператор <code>continue</code>	116
Проверяем, завершился ли цикл раньше, с помощью блока <code>else</code>	116
Генерируем числовые последовательности с помощью функции <code>range()</code>	117
Прочие итераторы	118
Читайте далее.....	118
Упражнения	118

Глава 7. Кортежи и списки	119
Кортежи	119
Создаем кортежи с помощью запятых и оператора ()	120
Создаем кортежи с помощью функции tuple()	121
Объединяем кортежи с помощью оператора +	121
Размножаем элементы с помощью оператора *	122
Сравниваем кортежи	122
Итерируем по кортежам с помощью for и in	122
Изменяем кортеж	122
Списки	123
Создаем списки с помощью скобок []	123
Создаем список или преобразуем в список с помощью функции list()	123
Создаем список из строки с использованием функции split()	124
Получаем элемент с помощью конструкции [смещение]	124
Извлекаем элементы с помощью разделения	125
Добавляем элемент в конец списка с помощью функции append()	126
Добавляем элемент на определенное место с помощью функции insert()	126
Размножаем элементы с помощью оператора *	127
Объединяем списки с помощью метода extend() или оператора +	127
Изменяем элемент с помощью конструкции [смещение]	128
Изменяем элементы с помощью разделения	128
Удаляем заданный элемент с помощью оператора del	129
Удаляем элемент по значению с помощью функции remove()	129
Получаем и удаляем заданный элемент с помощью функции pop()	129
Удаляем все элементы с помощью функции clear()	130
Определяем смещение по значению с помощью функции index()	130
Проверяем на наличие элемента в списке с помощью оператора in	131
Подсчитываем количество вclusions значения с помощью функции count()	131
Преобразуем список в строку с помощью функции join()	131
Меняем порядок элементов с помощью функций sort() или sorted()	132
Получаем длину списка с помощью функции len()	133
Присваиваем с помощью оператора =	133
Копируем списки с помощью функций copy() и list() или путем разделения	134
Копируем все с помощью функции deepcopy()	134
Сравниваем списки	135
Итерируем по спискам с помощью операторов for и in	136

Итерируем по нескольким последовательностям с помощью функции <code>zip()</code>	137
Создаем список с помощью списковых включений	138
Списки списков	140
Кортежи или списки?	141
Включений кортежей не существует	141
Читайте далее	142
Упражнения	142
Глава 8. Словари и множества	144
Словари	144
Создаем словарь с помощью <code>{}</code>	144
Создаем словарь с помощью функции <code>dict()</code>	145
Преобразуем с помощью функции <code>dict()</code>	146
Добавляем или изменяем элемент с помощью конструкции [ключ]	146
Получаем элемент словаря с помощью конструкции [ключ] или функции <code>get()</code>	148
Получаем все ключи с помощью функции <code>keys()</code>	148
Получаем все значения с помощью функции <code>values()</code>	149
Получаем все пары «ключ — значение» с помощью функции <code>items()</code>	149
Получаем длину словаря с помощью функции <code>len()</code>	149
Объединяем словари с помощью конструкции <code>{**a, **b}</code>	149
Объединяем словари с помощью функции <code>update()</code>	150
Удаляем элементы по их ключу с помощью оператора <code>del</code>	151
Получаем элемент по ключу и удаляем его с помощью функции <code>pop()</code>	151
Удаляем все элементы с помощью функции <code>clear()</code>	151
Проверяем на наличие ключа с помощью оператора <code>in</code>	152
Присваиваем значения с помощью оператора <code>=</code>	152
Копируем значения с помощью функции <code>copy()</code>	152
Копируем все с помощью функции <code>deepcopy()</code>	153
Сравниваем словари	154
Итерируем по словарям с помощью <code>for</code> и <code>in</code>	154
Включения словарей	155
Множества	156
Создаем множество с помощью функции <code>set()</code>	157
Преобразуем другие типы данных с помощью функции <code>set()</code>	157
Получаем длину множества с помощью функции <code>len()</code>	158
Добавляем элемент с помощью функции <code>add()</code>	158
Удаляем элемент с помощью функции <code>remove()</code>	158

Итерируем по множествам с помощью <code>for</code> и <code>in</code>	158
Проверяем на наличие значения с помощью оператора <code>in</code>	158
Комбинации и операторы	159
Включение множества.....	162
Создаем неизменяемое множество с помощью функции <code>frozenset()</code>	162
Структуры данных, которые мы уже рассмотрели	163
Создание крупных структур данных.....	164
Читайте далее.....	164
Упражнения.....	165
Глава 9. Функции	166
Определяем функцию с помощью ключевого слова <code>def</code>	166
Вызываем функцию с помощью скобок	167
Аргументы и параметры.....	167
None — это полезно.....	169
Позиционные аргументы	170
Аргументы — ключевые слова.....	171
Указываем значение параметра по умолчанию	171
Получаем/разбиваем аргументы — ключевые слова с помощью символа <code>*</code>	172
Получаем/разбиваем аргументы — ключевые слова с помощью символов <code>**</code>	174
Аргументы, передаваемые только по ключевым словам.....	175
Изменяемые и неизменяемые аргументы	176
Строки документации.....	176
Функции — это объекты первого класса.....	177
Внутренние функции	179
Анонимные функции: лямбда-выражения	181
Генераторы	182
Функции-генераторы	182
Включения генераторов.....	183
Декораторы	183
Пространства имен и область определения	186
Использование символов <code>_</code> и <code>__</code> в именах.....	188
Рекурсия	188
Асинхронные функции.....	190
Исключения	190
Обрабатываем ошибки с помощью операторов <code>try</code> и <code>except</code>	191
Создаем собственные исключения	192
Читайте далее.....	193
Упражнения.....	193

Глава 10. Ой-ой-ой: объекты и классы	194
Что такое объекты	194
Простые объекты	195
Определяем класс с помощью ключевого слова <code>class</code>	195
Атрибуты	196
Методы	197
Инициализация	197
Наследование	198
Наследование от родительского класса	199
Переопределение методов	200
Добавление метода	201
Получаем помощь от своего родителя с использованием метода <code>super()</code>	202
Множественное наследование	203
Примеси	205
В защиту <code>self</code>	205
Доступ к атрибутам	206
Прямой доступ	206
Геттеры и сеттеры	206
Свойства для доступа к атрибутам	207
Свойства для вычисляемых значений	209
Искажение имен для безопасности	209
Атрибуты классов и объектов	210
Типы методов	211
Методы объектов	211
Методы классов	212
Статические методы	212
Утиная типизация	213
Магические методы	215
Агрегирование и композиция	218
Когда использовать объекты, а когда — что-то другое	218
Именованные кортежи	219
Классы данных	221
<code>attrs</code>	222
Читайте далее	222
Упражнения	222
Глава 11. Модули, пакеты и программы	224
Модули и оператор <code>import</code>	224
Импортируем модуль	224

Импортируем модуль с другим именем.....	226
Импортируем только самое необходимое.....	226
Пакеты.....	227
Путь поиска модуля.....	228
Относительный и абсолютный импорт.....	229
Пакеты пространств имен.....	229
Модули против объектов.....	230
Достоинства стандартной библиотеки Python.....	231
Обрабатываем отсутствующие ключи с помощью функций setdefault() и defaultdict().....	231
Подсчитываем элементы с помощью функции Counter().....	233
Упорядочиваем по ключу с помощью OrderedDict().....	235
Стек + очередь == deque.....	235
Итерируем по структурам кода с помощью модуля itertools.....	236
Красиво выводим данные на экран с помощью функции pprint().....	238
Работаем со случайными числами.....	238
Нужно больше кода.....	239
Читайте далее.....	240
Упражнения.....	240

Часть II. Python на практике

Глава 12. Обрабатываем данные.....	242
Текстовые строки: Unicode.....	243
Строки формата Unicode в Python 3.....	244
Кодирование и декодирование с помощью кодировки UTF-8.....	246
Кодирование.....	247
Декодирование.....	249
Сущности HTML.....	250
Нормализация.....	251
Подробная информация.....	252
Текстовые строки: регулярные выражения.....	253
Ищем точное начальное совпадение с помощью функции match().....	254
Ищем первое совпадение с помощью функции search().....	255
Ищем все совпадения, используя функцию findall().....	255
Разбиваем совпадения с помощью функции split().....	256
Заменяем совпадения с помощью функции sub().....	256
Шаблоны: специальные символы.....	256
Шаблоны: использование спецификаторов.....	258
Шаблоны: указываем способ вывода совпадения.....	261

Бинарные данные	261
bytes и bytearray.....	262
Преобразуем бинарные данные с помощью модуля struct.....	263
Другие инструменты для работы с бинарными данными	266
Преобразуем байты/строки с помощью модуля binascii	267
Битовые операторы.....	267
Аналогия с ювелирными изделиями	268
Читайте далее.....	268
Упражнения	268
Глава 13. Календари и часы.....	271
Високосный год.....	272
Модуль datetime.....	273
Модуль time.....	275
Читаем и записываем дату и время	277
Все преобразования	281
Альтернативные модули	281
Читайте далее.....	282
Упражнения	282
Глава 14. Файлы и каталоги.....	283
Ввод информации в файлы и ее вывод из них	283
Создаем или открываем файлы с помощью функции open()	284
Записываем в текстовый файл с помощью функции print().....	284
Записываем в текстовый файл с помощью функции write()	285
Считываем данные из текстового файла, используя функции read(), readline() и readlines()	286
Записываем данные в бинарный файл с помощью функции write()	288
Читаем бинарные файлы с помощью функции read().....	289
Закрываем файлы автоматически с помощью ключевого слова with.....	289
Меняем позицию с помощью функции seek().....	289
Отображение в памяти	291
Операции с файлами	292
Проверяем существование файла с помощью функции exists()	292
Проверяем тип с помощью функции isfile().....	292
Копируем файлы, используя функцию copy().....	293
Изменяем имена файлов с помощью функции rename().....	293
Создаем ссылки с помощью функции link() или symlink().....	293
Изменяем разрешения с помощью функции chmod()	294
Изменение владельца файла с помощью функции chown().....	294
Удаляем файл с помощью функции remove().....	294

Каталоги.....	295
Создаем каталог с помощью функции <code>mkdir()</code>	295
Удаляем каталог, используя функцию <code>rmdir()</code>	295
Выводим на экран содержимое каталога с помощью функции <code>listdir()</code>	295
Изменяем текущий каталог с помощью функции <code>chdir()</code>	296
Перечисляем совпадающие файлы, используя функцию <code>glob()</code>	296
Pathname	297
Получаем путь с помощью функции <code>abspath()</code>	298
Получаем символическую ссылку с помощью функции <code>realpath()</code>	298
Построение пути с помощью функции <code>os.path.join()</code>	298
Модуль <code>pathlib</code>	298
BytesIO и StringIO.....	299
Читайте далее.....	301
Упражнения.....	301
Глава 15. Данные во времени: процессы и конкурентность	302
Программы и процессы	302
Создаем процесс с помощью модуля <code>subprocess</code>	303
Создаем процесс с помощью модуля <code>multiprocessing</code>	304
Убиваем процесс, используя функцию <code>terminate()</code>	305
Получаем системную информацию с помощью модуля <code>os</code>	306
Получаем информацию о процессах с помощью модуля <code>psutil</code>	306
Автоматизация команд.....	307
Invoke.....	307
Другие вспомогательные методы для команд.....	308
Конкурентность	308
Очереди	309
Процессы.....	310
Потоки	311
Concurrent.futures.....	314
Зеленые потоки и <code>gevent</code>	317
twisted	320
asyncio	321
Redis.....	321
Помимо очередей.....	325
Читайте далее.....	326
Упражнения.....	326
Глава 16. Данные в коробке: надежные хранилища	327
Плоские текстовые файлы	327
Текстовые файлы, дополненные пробелами	328

Структурированные текстовые файлы.....	328
CSV	328
XML	331
Примечание о безопасности XML	333
HTML.....	333
JSON.....	334
YAML.....	337
Tablib.....	338
Pandas	338
Конфигурационные файлы	340
Бинарные файлы	341
Электронные таблицы	341
HDF5	341
TileDB.....	342
Реляционные базы данных.....	342
SQL	343
DB-API	345
SQLite.....	345
MySQL.....	347
PostgreSQL.....	347
SQLAlchemy.....	348
Другие пакеты для работы с базами данных	354
Хранилища данных NoSQL	354
Семейство dbm.....	354
Memcached.....	355
Redis.....	356
Документоориентированные базы данных.....	363
Базы данных временных рядов.....	364
Графовые базы данных.....	365
Другие серверы NoSQL	365
Полнотекстовые базы данных.....	366
Читайте далее.....	366
Упражнения.....	366
Глава 17. Данные в пространстве: сети	368
TCP/IP.....	368
Сокеты	370
scapy	374
Netcat.....	374

Паттерны для работы с сетями.....	375
Паттерн «Запрос — ответ»	375
ZeroMQ.....	375
Другие инструменты обмена сообщениями	380
Паттерн «Публикация — подписка».....	380
Redis.....	380
ZeroMQ.....	382
Другие инструменты «Публикации — подписки»	383
Интернет-сервисы.....	384
Доменная система имен	384
Модули Python для работы с электронной почтой	385
Другие протоколы	385
Веб-сервисы и API	385
Сериализация данных.....	386
Сериализация с помощью pickle.....	387
Другие форматы сериализации.....	388
Удаленные вызовы процедур.....	388
XML RPC	389
JSON RPC.....	390
MessagePack RPC.....	391
Zerorpc.....	392
gRPC	393
Twirp	393
Инструменты удаленного управления.....	394
Работаем с большими объемами данных	394
Hadoop	394
Spark	395
Disco.....	395
Dask.....	395
Работаем в облаках	396
Amazon Web Services.....	397
Google.....	397
Microsoft Azure.....	397
OpenStack	398
Docker.....	398
Kubernetes.....	398
Читайте далее.....	398
Упражнения.....	399

Глава 18. Распутываем Всемирную паутину	400
Веб-клиенты	401
Тестируем с помощью telnet.....	402
Тестируем с помощью curl	403
Тестируем с использованием httpie.....	404
Тестируем с помощью httpbin.....	405
Стандартные веб-библиотеки Python.....	405
За пределами стандартной библиотеки: requests.....	407
Веб-серверы.....	408
Простейший веб-сервер Python.....	409
Web Server Gateway Interface (WSGI)	410
ASGI	411
apache.....	411
NGINX	412
Другие веб-серверы Python.....	413
Фреймворки для работы веб-серверами	413
Bottle	414
Flask	416
Django.....	420
Другие фреймворки	421
Фреймворки для работы с базами данных	421
Веб-сервисы и автоматизация.....	422
Модуль webbrowser.....	422
Модуль webview	423
REST API	424
Поиск и выборка данных	424
Scrapy	425
BeautifulSoup	425
Requests-HTML.....	426
Давайте посмотрим фильм.....	426
Читайте далее.....	429
Упражнения	429
Глава 19. Быть питонщиком	431
О программировании	431
Ищем код на Python	432
Установка пакетов	432
pip	433
virtualenv	434

pipenv.....	434
Менеджер пакетов	434
Установка из исходного кода	435
Интегрированные среды разработки.....	435
IDLE	435
PyCharm	435
IPython.....	436
Jupyter Notebook.....	438
JupyterLab.....	438
Именованное и документирование	438
Добавление подсказок типов.....	440
Тестирование кода.....	440
Программы pylint, pyflakes, flake8 или PEP-8.....	441
Пакет unittest.....	443
Пакет doctest.....	447
Пакет nose.....	448
Другие фреймворки для тестирования	449
Постоянная интеграция.....	449
Отладка кода	450
Функция print()	450
Отладка с помощью декораторов.....	451
Отладчик pdb.....	452
Функция breakpoint()	458
Записываем в журнал сообщения об ошибках.....	458
Оптимизация кода	460
Измеряем время	461
Алгоритмы и структуры данных	464
Cython, NumPy и расширения C	465
PyPy.....	465
Numba.....	466
Управление исходным кодом	467
Mercurial.....	467
Git	467
Распространение ваших программ.....	470
Клонируйте эту книгу.....	470
Как узнать больше.....	470
Книги.....	471
Сайты	471
Группы.....	472

Конференции.....	472
Вакансии, связанные с Python	472
Читайте далее.....	473
Упражнения.....	473
Глава 20. Пи-Арт.....	474
Двумерная графика.....	474
Стандартная библиотека.....	474
PIL и Pillow	475
ImageMagick.....	478
Трехмерная графика	478
Трехмерная анимация	479
Графические пользовательские интерфейсы (GUI).....	479
Диаграммы, графики и визуализация.....	481
Matplotlib.....	481
Seaborn	483
Bokeh	485
Игры.....	485
Аудио и музыка	486
Читайте далее.....	486
Упражнения.....	486
Глава 21. За работой.....	487
The Microsoft Office Suite	487
Выполняем бизнес-задачи.....	488
Обработка бизнес-данных.....	489
Извлечение, преобразование и загрузка.....	489
Валидация данных.....	493
Дополнительные источники информации.....	493
Пакеты для работы с бизнес-данными с открытым исходным кодом	494
Python в области финансов.....	494
Безопасность бизнес-данных	495
Карты	495
Форматы	496
Нарисуем карту на основе шейп-файла.....	496
Geopandas.....	498
Другие пакеты для работы с картами	500
Приложения и данные	501
Читайте далее.....	502
Упражнения.....	502

Глава 22. Python в науке.....	503
Математика и статистика в стандартной библиотеке	503
Математические функции	503
Работа с комплексными числами.....	505
Рассчитываем точное значение чисел с плавающей точкой с помощью модуля decimal	506
Выполняем вычисления для рациональных чисел с помощью модуля fractions	507
Используем Packed Sequences с помощью модуля array.....	507
Обрабатываем простую статистику с помощью модуля statistics	508
Перемножение матриц	508
Python для науки.....	508
NumPy.....	508
Создаем массив с помощью функции array()	509
Создаем массив с помощью функции arange().....	510
Создаем массив с помощью функций zeros(), ones() и random().....	511
Изменяем форму массива с помощью метода reshape()	512
Получаем элемент с помощью конструкции []	513
Математика массивов.....	514
Линейная алгебра.....	514
Библиотека SciPy	515
Библиотека SciKit	516
Pandas.....	516
Python и научные области.....	517
Читайте далее.....	518
Упражнения.....	518

Приложения

Приложение А. Аппаратное и программное обеспечение для начинающих программистов	520
Аппаратное обеспечение	520
Компьютеры пещерных людей.....	520
Электричество.....	521
Изобретения	521
Идеальный компьютер.....	522
Процессор.....	522
Память и кэш.....	522
Хранение	522
Ввод данных.....	523

Вывод данных.....	523
Относительное время доступа.....	523
Программное обеспечение	524
Вначале был бит	524
Машинный язык.....	524
Ассемблер	525
Высокоуровневые языки.....	525
Операционные системы.....	526
Виртуальные машины	527
Контейнеры.....	527
Распределенные вычисления и сети	527
Облако	528
Kubernetes.....	528
Приложение Б. Установка Python 3	530
Проверьте свою версию Python.....	530
Установка стандартной версии Python	531
macOS.....	532
Windows.....	534
Linux или Unix	535
Установка менеджера пакетов pip	535
Установка virtualenv	535
Другие способы работы с пакетами	536
Устанавливаем Anaconda	536
Приложение В. Нечто совершенно иное: async	538
Сопрограммы и циклы событий.....	538
async против... ..	542
Асинхронные фреймворки и серверы	542
Приложение Г. Ответы к упражнениям	544
1. Python: с чем его едят	544
2. Типы данных, значения, переменные и имена	545
3. Числа	545
4. Выбираем с помощью if.....	546
5. Текстовые строки	547
6. Создаем циклы с помощью ключевых слов while и for.....	551
7. Кортежи и списки	552
8. Словари и множества.....	556
9. Функции.....	559
10. Ой-ой-ой: объекты и классы	560

11. Модули, пакеты и программы	564
12. Обрабатываем данные	566
13. Календари и часы	571
14. Файлы и каталоги	572
15. Данные во времени: процессы и конкурентность.....	573
16. Данные в коробке: устойчивые хранилища	574
17. Данные в пространстве: сети.....	577
18. Распутываем Всемирную паутину	584
19. Быть питонщиком.....	585
20. Пи-Арт	585
21. За работой.....	586
22. Python в науке	586
Приложение Д. Вспомогательные таблицы	587
Приоритет операторов.....	587
Строковые методы	588
Изменение регистра	588
Поиск.....	588
Изменение	588
Форматирование	589
Тип строки	589
Атрибуты модуля string	589
Эпилог.....	591

*С любовью к Мэри, Тому и Рокси,
а также Кэрин и Эрику.*

Введение

Как и обещает название, книга познакомит вас с одним из самых популярных языков программирования — Python. Издание предназначено как для начинающих программистов, так и для тех, кто уже имеет опыт в написании программ и просто желает добавить Python к списку доступных ему языков.

В большинстве случаев изучать компьютерный язык проще, чем человеческий, — в нем меньше двусмысленностей и исключений, которые приходится запоминать. Python — один из самых последовательных и понятных компьютерных языков, он сочетает в себе простоту изучения, простоту использования и большую выразительную силу.

Компьютерные языки состоят из *данных* — аналогами в разговорной речи являются существительные — и *инструкций* (или *кода*), которые можно сравнить с глаголами. Изучить нужно будет и то и другое: вы освоите основы кода и структур данных и узнаете, как их объединить. Затем перейдете к более сложным темам, а программы, которые вы будете читать и писать, станут длиннее и сложнее. Если провести аналогию с работой по дереву, мы начнем с молотка, гвоздей и небольших кусков древесины, а во второй половине книги обратимся к более специализированным инструментам, которые можно сравнить с токарными станками и другими более сложными устройствами.

Вам нужно знать не только сам язык, но и то, что с его помощью можно делать. Мы начнем с языка Python и его стандартной библиотеки, готовой к работе прямо «из коробки». Помимо этого, я покажу вам, как находить, загружать, устанавливать и использовать качественные сторонние пакеты: касаться узких тем или рассматривать сложные трюки не буду, а сделаю акцент на том, что после десяти лет работы с Python считаю действительно полезным.

Несмотря на то что книга представляет собой введение в Python, в ней мы затронем и несколько дополнительных тем, с которыми, на мой взгляд, следует ознакомиться еще на начальном этапе. Работе с базами данных и Интернетом мы тоже уделим внимание, однако не станем забывать, что технологии меняются очень быстро — теперь от программиста на Python общество ждет знаний об облачных технологиях, машинном обучении и создании потоков событий: основную информацию по этим темам вы также здесь найдете.

Язык Python имеет некоторые специальные функции, работающие лучше, чем адаптированные стили программирования из других языков. Например, использование ключевого слова `for` и *итераторов* — более прямой способ создания

цикла: пользоваться им гораздо удобнее, нежели вручную инкрементировать переменную-счетчик.

Когда вы изучаете что-то новое, бывает трудно определить с тем, какие слова являются терминами и какие понятия на самом деле важны. Иначе говоря, тестировалась ли эта функциональность? Я выделю некоторые термины и понятия, которые имеют особое значение в Python. Код, написанный на языке Python, можно будет увидеть даже в самых первых главах.



Подобные примечания я буду делать, когда что-то может быть непонятным или же существует более питонский способ решить проблему.

Python неидеален. Я обращаю ваше внимание на то, что кажется сомнительным и чего следует избегать, и предложу альтернативные варианты.

По некоторым темам, таким как наследование объектов, MVC или проектирование REST для работы с Интернетом, мое мнение может отличаться от общепринятого. Вам самим решать, к кому прислушаться.

Для кого эта книга

Эта книга для всех, кто заинтересован в изучении одного из самых популярных во всем мире языков программирования. Наличие или отсутствие опыта с другими языками программирования не имеет значения.

Что нового во втором издании

Что изменилось со времени выхода первого издания?

- ☐ Добавилось около 100 страниц, в том числе с изображением котиков.
- ☐ Количество глав удвоилось, но сами главы стали короче.
- ☐ В начале книги появилась глава, посвященная типам данных, переменным и именам.
- ☐ Рассмотрены новые особенности Python, такие как *f-строки*.
- ☐ Рекомендованы новые или улучшенные сторонние библиотеки.
- ☐ Во всей книге присутствуют новые примеры кода.
- ☐ Для начинающих разработчиков добавлен текст с описанием аппаратного и программного обеспечения.
- ☐ Более опытные разработчики могут ознакомиться с библиотекой *asyncio*.
- ☐ Рассмотрен новый стек технологий: контейнеры, облачные технологии, наука о данных и машинное обучение.

- ❑ Добавлены подсказки, с помощью которых вы сможете найти работу программиста на Python.

Что не изменилось? Примеры, в которых используются плохие стихотворения и утки. Они с нами навсегда.

Структура книги

В первой части излагаются основы языка программирования Python: главы 1–11 следует читать по порядку. Я оперирую простейшими структурами данных и кода, постепенно составляя из них более сложные и реалистичные программы. Во второй части (главы 12–22) показывается, каким образом язык программирования Python используется в определенных прикладных областях, таких как Интернет, базы данных, сети и т. д.: эти главы можно читать в любом порядке.

Вот краткое содержание всех глав и приложений и обзор новых терминов, с которыми вы там встретитесь.

- ❑ *Глава 1. «Python: с чем его едят».* Компьютерные программы не так уж и отличаются от других инструкций, с которыми вы сталкиваетесь каждый день. Мы рассмотрим небольшие программы, написанные на Python. Они продемонстрируют синтаксис языка, его возможности и способы применения в реальном мире. Вы узнаете, как запустить программу внутри *интерактивного интерпретатора (оболочки)*, а также из текстового *файла*, сохраненного на вашем компьютере.
- ❑ *Глава 2. «Данные: типы, значения, переменные и имена».* В компьютерных языках используются данные и инструкции. Компьютер по-разному хранит и обрабатывает разные *типы* данных. Их значения или можно изменять (такие типы называются *изменяемыми*), или нельзя (*неизменяемые* типы). В программе, написанной на Python, данные могут быть представлены как литералами (числами вроде 78 или текстовыми *строками* вроде "waffle"), так и именованными *переменными*. В отличие от многих других языков программирования Python относится к переменным как к *именам*, и это влечет за собой некоторые важные последствия.
- ❑ *Глава 3. «Числа».* В этой главе показываются простейшие типы данных, применяемые в языке программирования Python: *булевы переменные*, *целые числа* и *числа с плавающей точкой*. Вы также изучите простейшую математику. В примерах этой главы интерактивный интерпретатор Python используется как калькулятор.
- ❑ *Глава 4. «Выбираем с помощью оператора if».* С существительными (типами данных) и с глаголами (программными структурами) мы поработаем в нескольких главах. Код, написанный на Python, обычно выполняется по одной строке за раз: от начала программы до ее конца. Структура *if* позволяет запускать разные строки кода исходя из результата сравнения определенных данных.
- ❑ *Глава 5. «Текстовые строки».* Здесь мы обратимся к существительным и миру текстовых *строк*. Вы научитесь создавать, объединять, изменять и получать строки, а также выводить их на экран.

- ❑ *Глава 6. «Создаем циклы с помощью ключевых слов `while` и `for`».* Снова глаголы. Вы научитесь создавать *цикл* двумя способами — с помощью `for` и с помощью `while`, а также узнаете, что такое *итераторы* — одно из основных понятий Python.
- ❑ *Глава 7. «Кортежи и списки».* Пришло время рассмотреть первые структуры данных более высокого уровня: списки и кортежи. Они представляют собой последовательности значений, которыми вы будете пользоваться как конструктором Lego для того, чтобы создавать более сложные структуры. Вы научитесь *проходить* по ним с помощью *итераторов*, а также быстро создавать списки с помощью *списковых включений*.
- ❑ *Глава 8. «Словари и множества».* Словари и множества позволяют сохранять данные не по позиции, а по их значению. Вы увидите, насколько это удобно, — данная особенность Python станет одной из ваших любимых.
- ❑ *Глава 9. «Функции».* Соединяйте структуры данных из предыдущих глав со структурами кода, чтобы выполнять сравнение, выборку или повторение операций. Упаковывайте код в функции и обрабатывайте ошибки с помощью *исключений*.
- ❑ *Глава 10. «Ой-ой-ой: объекты и классы».* Слово «объект» недостаточно конкретно, но имеет большое значение во многих компьютерных языках, в том числе и в Python. Если вы уже занимались объектно-ориентированным программированием на других языках, то в сравнении с ними Python покажется вам более простым. В этой главе объясняется, когда следует использовать объекты и классы, а когда лучше выбрать другой путь.
- ❑ *Глава 11. «Модули, пакеты и программы».* Вы узнаете, как перейти к более крупным структурам кода — *модулям*, *пакетам* и *программам*, а также где можно разместить код и данные, как ввести и вывести данные, обработать различные параметры, просмотреть стандартную библиотеку Python и то, что находится вне ее.
- ❑ *Глава 12. «Обрабатываем данные».* Вы научитесь профессионально обрабатывать данные и управлять ими. Эта глава полностью посвящена текстовым и двоичным данным, особенностям использования символов стандарта Unicode, а также поиску текста с помощью *регулярных* выражений. Вы познакомитесь с типами данных `byte` и `bytearray` — соперниками типа `string`, в которых содержатся необработанные бинарные значения вместо текстовых символов.
- ❑ *Глава 13. «Календари и часы».* С датой и временем работать бывает непросто. Здесь мы рассмотрим распространенные проблемы и способы их решения.
- ❑ *Глава 14. «Файлы и каталоги».* Простые хранилища данных используют *файлы* и *каталоги*. В этой главе речь пойдет о создании и использовании файлов и каталогов.
- ❑ *Глава 15. «Данные во времени: процессы и конкурентность».* Это первая глава, в которой мы приступаем к изучению системы. Начнем с данных во времени — вы научитесь использовать *программы*, *процессы* и *потoki* для того, чтобы выполнять больше работы за один промежуток времени (*конкурентность*). Среди прочего будут упомянуты последние добавления в *async* (более подробно они рассматриваются в приложении В).

- ❑ *Глава 16. «Данные в коробке: надежные хранилища».* Данные могут храниться в простых файлах и каталогах внутри файловых систем и структурироваться с помощью распространенных форматов, таких как CSV, JSON и XML. Однако по мере того, как объем и сложность данных будут расти, вам, возможно, придется использовать *базы данных* — как традиционные *реляционные*, так и современные базы данных *NoSQL*.
- ❑ *Глава 17. «Данные в пространстве: сети».* Отправляйте ваш код и данные через пространство по *сетям* с помощью *служб, протоколов* и *API*. В качестве примеров рассматриваются как низкоуровневые *сокеты*, библиотеки *обмена сообщениями* и системы массового обслуживания, так и развертывание в *облачных* системах.
- ❑ *Глава 18. «Распутываем Всемирную паутину».* Всемирной сети посвящена отдельная глава, в которой рассматриваются клиенты, серверы, извлечение данных, API и фреймворки. Вы научитесь *искать* сайты и *извлекать* из них данные, а затем разработаете реальный сайт, используя параметры *запросов* и *шаблоны*.
- ❑ *Глава 19. «Быть питонщиком».* В этой главе содержатся советы для программистов, пишущих на Python: вы получите рекомендации по установке (с помощью *pip* и *virtualenv*), использованию IDE, тестированию, отладке, журналированию, контролю исходного кода и документации. Узнаете также, как найти и установить полезные пакеты сторонних разработчиков, как упаковать свой код для повторного использования и где получить более подробную информацию.
- ❑ *Глава 20. «Пи-Арт».* При помощи языка программирования Python можно создавать произведения искусства: в графике, музыке, анимации и играх.
- ❑ *Глава 21. «За работой».* У Python есть специальные приложения для бизнеса: визуализация данных (графики, графы и карты), безопасность и регулирование.
- ❑ *Глава 22. «Python в науке».* За последние несколько лет Python стал главным языком науки, он используется в математике, статистике, физике, биологии и медицине. Его сильные стороны — *наука о данных* и *машинное обучение*. В этой главе демонстрируются возможности таких инструментов, как NumPy, SciPy и Pandas.
- ❑ *Приложение А. «Аппаратное и программное обеспечение для начинающих программистов».* Если вы новичок в мире программирования, из этого приложения вы можете узнать, как на самом деле работает аппаратное и программное обеспечение и что означают некоторые термины, с которыми в дальнейшем вам придется сталкиваться.
- ❑ *Приложение Б. «Установка Python 3».* Если вы еще не установили Python 3 на свой компьютер, в этом приложении вы найдете информацию о том, как это сделать независимо от того, какая операционная система у вас установлена: Windows, Mac OS/X, Linux или другой вариант Unix.
- ❑ *Приложение В. «Нечто совершенно иное: async».* В разных релизах Python добавляется функциональность для работы с асинхронностью — разобраться с ней может быть сложно. Я упоминаю о ней в тех главах, в которых заходит речь об асинхронности, но в этом приложении рассматриваю тему более подробно.
- ❑ *Приложение Г. «Ответы к упражнениям».* Здесь содержатся ответы на упражнения, приведенные в конце каждой главы. Не подглядывайте туда, пока не по-

попытайтесь решить задачи самостоятельно, в противном случае вы рискуете превратиться в козленочка.

- *Приложение Д. «Вспомогательные таблицы».* В этом приложении содержатся справочные данные.

Версии Python

Языки программирования со временем изменяются — разработчики добавляют в них новые возможности и исправляют ошибки. Примеры этой книги написаны и протестированы для версии Python 3.7. Версия 3.7 является наиболее современной на момент выхода этой книги, и о самых значимых нововведениях я расскажу. Версия 3.8 вышла в конце 2019 года — я рассмотрю самую ожидаемую функциональность¹. Узнать, что и когда было добавлено в язык программирования Python, можно, посетив страницу <https://docs.python.org/3/whatsnew/>: там представлена техническая информация. Она, скорее всего, покажется трудной для понимания, если вы только начинаете изучать Python, но может пригодиться в будущем, если вам нужно будет писать программы для компьютеров, на которых установлены другие версии Python.

Условные обозначения

В этой книге приняты следующие шрифтовые соглашения.

Курсив

Им обозначаются новые термины и понятия.

Моноширинный шрифт

Используется в листингах программного кода, а также для имен и расширений файлов, названий путей, имен функций, команд, баз данных, переменных, операторов и ключевых слов.

Курсивный моноширинный шрифт

Указывает текст, который необходимо заменить пользовательскими значениями или значениями, определяемыми контекстом.



Так оформлены совет, предложение или замечание.



Таким образом оформлено предупреждение.

¹ Оригинальное издание выпущено до выхода версии 3.8. Текущая версия — 3.8.2. — *Примеч. ред.*

Использование примеров кода

Примеры кода и упражнения, приведенные в тексте, доступны для загрузки по адресу <https://github.com/madscheme/introducing-python>. Эта книга написана, чтобы помочь вам в работе: вы можете применить код, содержащийся в ней, в ваших программах и документации и не связываться с нами, чтобы спросить разрешения, если собираетесь воспользоваться небольшим фрагментом кода. Например, если вы пишете программу и кое-где вставляете в нее код из книги, никакого особого разрешения не требуется. Однако если вы запишете на диск примеры из книги и начнете раздавать или продавать такие диски, разрешение на это получить необходимо. Если вы цитируете это издание, отвечая на вопрос, или воспроизводите код из него в качестве примера, разрешения не требуется. Но если включаете значительный фрагмент кода из данной книги в документацию по вашему продукту, необходимо разрешение.

Ссылки на источник приветствуются, но не обязательны. В такие ссылки обычно включаются название книги, имя ее автора, название издательства и номер ISBN. Например: «Простой Python. Современный стиль программирования. 2-е изд. Билл Любанович. Питер, 2020. 978-5-4461-1639-3».

При любых сомнениях относительно превышения разрешенного объема использования примеров кода, приведенных в данной книге, можете обращаться к нам по адресу permissions@oreilly.com.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

Благодарности

Искренне благодарю обозревателей и читателей, помогавших сделать эту книгу лучше: Корбина Коллинза, Чарльза Гивра, Нейтана Стокса, Дейва Джорджа и Майка Джеймса.

Об авторе

Билл Любанович программировал в операционной системе Unix с 1977 года, разрабатывал GUI с 1981 года, базы данных — с 1990 года, а веб-разработкой занимался с 1993 года. Кроме того:

- ❑ 1982–1988 (Intran) — разрабатывал приложение Metaform на первой коммерческой графической рабочей станции;
- ❑ 1990–1995 (Northwest Airlines) — написал визуальную систему управления доходами, а также первый тест для анализа онлайн-маркетинга в Сети;
- ❑ 1994 (Tela) — стал сооснователем одного из первых интернет-провайдеров;
- ❑ 1995–1999 (WAM!NET) — разрабатывал веб-доски сообщений и 3M Digital Media Repository;
- ❑ 1999–2005 (Mad Scheme) — стал сооснователем компании, занимающейся веб-разработкой и хостингом;
- ❑ 2005 (O'Reilly) — в соавторстве написал несколько глав книги *Linux Server Security*¹;
- ❑ 2007 (O'Reilly) — в соавторстве написал книгу *Linux System Administration*²;
- ❑ 2010–2013 (Keep) — разработал и построил службы Core Services, соединяющие веб-фронтенды и бэкенды, работающие с базами данных;
- ❑ 2014 (O'Reilly) — написал книгу *Introducing Python* (первое издание)³;
- ❑ 2015–2016 (Internet Archive) — работал над API и адаптацией Wayback Machine на Python;
- ❑ 2016–2018 (CrowdStrike) — управлял сервисами на базе Python, которые обрабатывают миллиарды ежедневных событий, связанных с безопасностью.

Билл счастливо живет в горах Сангре-де-Сасквоч в штате Миннесота со своей чудесной семьей: женой Мэри, сыном Томом (и его женой Рокси) и дочерью Кэрин (и ее мужем Эриком), ухаживает за кошками Ингой и Люси и котом Честером.

¹ *Bauer M. Linux Server Security, 2nd Edition. — O'Reilly, 2009.*

² *Адельштайн Т., Любанович Б. Системное администрирование в Linux. — СПб.: Питер, 2010.*

³ *Любанович Б. Простой Python. Современный стиль программирования. — СПб.: Питер, 2019.*