

Оглавление

Введение	23
Аудитория	23
Краткое описание	23
Версии Python	25
Соглашения, принятые в этой книге.	26
Использование примеров кода	26
Как с нами связаться	27
Благодарности	27
Глава 1. Python: с чем его едят	28
Python в реальном мире.	33
Python против языка X	33
Почему же Python?	37
Когда не стоит использовать Python.	37
Python 2 против Python 3	38
Установка Python	39
Запуск Python	39
Использование интерактивного интерпретатора	39
Использование файлов Python.	40
Что дальше?	41
Момент просветления	41
Упражнения.	42

Глава 2. Ингредиенты Python: числа, строки и переменные	43
Переменные, имена и объекты.	43
Числа	47
Целые числа	47
Приоритет операций	51
Системы счисления	52
Преобразования типов	53
Насколько объем тип int?	55
Числа с плавающей точкой	55
Математические функции	56
Строки.	56
Создаем строки с помощью кавычек	57
Преобразование типов данных с помощью функции str()	59
Создаем управляющие символы с помощью символа \	60
Объединяем строки с помощью символа +	61
Размножаем строки с помощью символа *	61
Извлекаем символ с помощью символов []	61
Извлекаем подстроки с помощью оператора [start : end : step]	62
Получаем длину строки с помощью функции len()	65
Разделяем строку с помощью функции split()	65
Объединяем строки с помощью функции join()	66
Развлекаемся со строками	66
Регистр и выравнивание	67
Заменяем символы с помощью функции replace()	68
Больше действий со строками	69
Упражнения.	69
Глава 3. Наполнение Python: списки, кортежи, словари и множества	70
Списки и кортежи	70
Списки.	71
Создание списков с помощью оператора [] или метода list()	71
Преобразование других типов данных в списки с помощью функции list()	71

Получение элемента с помощью конструкции [смещение]	72
Списки списков	73
Изменение элемента с помощью конструкции [смещение]	74
Отрежьте кусочек — извлечение элементов с помощью диапазона смещений	74
Добавление элемента в конец списка с помощью метода append() . . .	75
Объединяем списки с помощью метода extend() или оператора +=	75
Добавление элемента с помощью функции insert()	76
Удаление заданного элемента с помощью функции del	76
Удаление элемента по значению с помощью функции remove()	77
Получение заданного элемента и его удаление с помощью функции pop()	77
Определение смещения элемента по значению с помощью функции index()	77
Проверка на наличие элемента в списке с помощью оператора in . . .	78
Определяем количество включений значения с помощью функции count()	78
Преобразование списка в строку с помощью функции join()	78
Меняем порядок элементов с помощью функции sort()	79
Получение длины списка с помощью функции len()	80
Присваивание с помощью оператора =, копирование с помощью функции copy()	80
Кортежи	81
Создание кортежей с помощью оператора ()	82
Кортежи против списков	83
Словари	83
Создание словаря с помощью {}	84
Преобразование с помощью функции dict()	84
Добавление или изменение элемента с помощью конструкции [ключ]	85
Объединение словарей с помощью функции update()	87
Удаление элементов по их ключу с помощью del	87
Удаление всех элементов с помощью функции clear()	88
Проверяем на наличие ключа с помощью in	88

Получение элемента словаря с помощью конструкции [ключ]	89
Получение всех ключей с помощью функции keys()	89
Получение всех значений с помощью функции values()	90
Получение всех пар «ключ — значение» с помощью функции items()	90
Присваиваем значения с помощью оператора =, копируем их с помощью функции copy()	90
Множества.	91
Создание множества с помощью функции set()	92
Преобразование других типов данных с помощью функции set()	92
Проверяем на наличие значения с помощью ключевого слова in	93
Комбинации и операторы	94
Сравнение структур данных.	97
Создание крупных структур данных	97
Упражнения.	98
Глава 4. Корочка Python: структуры кода	100
Комментируем с помощью символа #	100
Продлеваем строки с помощью символа \	101
Сравниваем выражения с помощью операторов if, elif и else.	102
Повторяем действия с помощью while	106
Прерываем цикл с помощью break	107
Пропускаем итерации с помощью continue.	107
Проверяем, завершился ли цикл заранее, с помощью else	108
Выполняем итерации с помощью for.	108
Прерываем цикл с помощью break.	110
Пропускаем итерации с помощью continue.	110
Проверяем, завершился ли цикл заранее, с помощью else	110
Итерирование по нескольким последовательностям с помощью функции zip()	111
Генерирование числовых последовательностей с помощью функции range()	112
Прочие итераторы.	113

Включения	113
Включение списков	113
Включение словаря	116
Включение множества	116
Включение генератора	117
Функции	118
Позиционные аргументы	122
Аргументы — ключевые слова	122
Указываем значение параметра по умолчанию	122
Получаем позиционные аргументы с помощью *	124
Получение аргументов — ключевых слов с помощью **	125
Строки документации	125
Функции — это граждане первого класса	126
Внутренние функции	128
Замыкания	129
Анонимные функции: функция lambda()	130
Генераторы	131
Декораторы	132
Пространства имен и область определения	134
Обработка ошибок с помощью try и except	137
Создание собственных исключений	139
Упражнения	140
Глава 5. Py Boxes: модули, пакеты и программы	142
Отдельные программы	142
Аргументы командной строки	143
Модули и утверждение import	143
Импортируем модуль	143
Импортируем модуль с другим именем	145
Импортируем только самое необходимое	145
Директории поиска модулей	146
Пакеты	146

Стандартная библиотека Python	147
Обработка отсутствующих ключей с помощью функций <code>setdefault()</code> и <code>defaultdict()</code>	148
Подсчитываем элементы с помощью функции <code>Counter()</code>	150
Упорядочиваем по ключу с помощью <code>OrderedDict()</code>	151
Стек + очередь == <code>deque</code>	152
Итерируем по структурам кода с помощью <code>itertools</code>	153
Выводим данные на экран красиво с помощью функции <code>pprint()</code>	155
Нужно больше кода	155
Упражнения	156
Глава 6. Ой-ой-ой: объекты и классы	157
Что такое объекты	157
Определяем класс с помощью ключевого слова <code>class</code>	158
Наследование	160
Перегрузка метода	161
Добавление метода	162
Просим помощи у родителя с помощью ключевого слова <code>super</code>	163
В защиту <code>self</code>	164
Получаем и устанавливаем значение атрибутов с помощью свойств	165
Искажение имен для безопасности	168
Типы методов	169
Утиная типизация	170
Особые методы	172
Композиция	175
Когда лучше использовать классы и объекты, а когда — модули	176
Упражнения	178
Глава 7. Работаем с данными профессионально	180
Текстовые строки	180
Unicode	180
Формат	188
Совпадение с регулярными выражениями	192

Бинарные данные	200
bytes и bytearray	200
Преобразуем бинарные данные с помощью модуля struct.	202
Другие инструменты для работы с бинарными данными.	205
Преобразование байтов/строк с помощью функции binascii().	206
Битовые операторы.	206
Упражнения.	207
Глава 8. Данные должны куда-то попадать.	210
Ввод информации в файлы и ее вывод из них	210
Запись в текстовый файл с помощью функции write()	211
Считываем данные из текстового файла с помощью функций read(), readline() и readlines()	213
Записываем данные в бинарный файл с помощью функции write().	215
Читаем бинарные файлы с помощью функции read()	216
Закрываем файлы автоматически с помощью ключевого слова with.	216
Меняем позицию с помощью функции seek()	216
Структурированные текстовые файлы	218
CSV.	219
XML.	221
HTML	223
JSON.	223
YAML.	226
Безопасность.	228
Конфигурационные файлы	228
Другие форматы обмена данными	229
Сериализация с помощью pickle.	230
Структурированные бинарные файлы	231
Электронные таблицы	231
HDF5.	231
Реляционные базы данных.	232
SQL.	233
DB-API.	234

SQLite	234
MySQL	236
PostgreSQL	237
SQLAlchemy	237
Хранилища данных NoSQL	244
Семейство dbm	244
Memcached	245
Redis	246
Прочие серверы NoSQL	254
Full-Text Databases	255
Упражнения	255
Глава 9. Распутываем Всемирную паутину.	257
Веб-клиенты	258
Тестируем с telnet	259
Стандартные веб-библиотеки Python	260
За пределами стандартной библиотеки: requests	262
Веб-серверы	263
Простейший веб-сервер Python	263
Web Server Gateway Interface	265
Фреймворки	265
Bottle	266
Flask	268
Веб-серверы, не использующие Python	272
Другие фреймворки	274
Веб-сервисы и автоматизация	276
Модуль webbrowser	276
API для Сети и Representational State Transfer	277
JSON	278
Поиск и выборка данных	278
Получаем HTML-код с помощью BeautifulSoup	278
Упражнения	280

Глава 10. Системы.	281
Файлы.	281
Создаем файл с помощью функции <code>open()</code>	281
Проверяем существование файла с помощью функции <code>exists()</code>	282
Проверяем тип с помощью функции <code>isfile()</code>	282
Копируем файлы с помощью функции <code>copy()</code>	283
Изменяем имена файлов с помощью функции <code>rename()</code>	283
Создаем ссылки с помощью <code>link()</code> или <code>symlink()</code>	283
Изменяем разрешения с помощью функции <code>chmod()</code>	284
Изменение владельца файла с помощью функции <code>chown()</code>	284
Получаем <code>pathname</code> с помощью функции <code>abspath()</code>	285
Получаем символическую ссылку с помощью функции <code>realpath()</code>	285
Удаляем файл с помощью функции <code>remove()</code>	285
Директории.	285
Создаем директорию с помощью функции <code>mkdir()</code>	285
Удаляем директорию с помощью функции <code>rmdir()</code>	286
Выведем на экран содержимое директории с помощью функции <code>listdir()</code>	286
Изменяем текущую директорию с помощью функции <code>chdir()</code>	287
Перечислим совпадающие файлы с помощью функции <code>glob()</code>	287
Программы и процессы.	287
Создаем процесс с помощью модуля <code>subprocess</code>	288
Создание процесса с помощью модуля <code>multiprocessing</code>	290
Убиваем процесс с помощью функции <code>terminate()</code>	290
Календари и часы.	291
Модуль <code>datetime</code>	292
Использование модуля <code>time</code>	295
Читаем и записываем дату и время	297
Альтернативные модули	300
Упражнения.	300

Глава 11. Конкуренция и сети	302
Конкуренция	303
Очереди	304
Процессы	305
Потоки	306
Зеленые потоки и <code>gevent</code>	308
<code>twisted</code>	311
<code>asyncio</code>	312
Redis	313
Помимо очередей	316
Сети	317
Шаблоны	317
Модель публикации-подписки	318
TCP/IP	322
Сокеты	323
ZeroMQ	327
Scapy	331
Интернет-службы	332
Веб-службы и API	334
Удаленная обработка	335
Большие данные и MapReduce	340
Работаем в облаках	341
Упражнения	344
Глава 12. Быть питонщиком	346
О программировании	346
Ищем код на Python	347
Установка пакетов	348
Используем <code>pip</code>	348
Использование менеджера пакетов	349
Установка из исходного кода	349
Интегрированные среды разработки	349
IDLE	350
PyCharm	350
IPython	350

Именуйте и документируйте.	351
Тестируем ваш код	352
Тестируем с помощью pylint, pyflakes и PEP-8.	352
Тестируем код с помощью unittest	354
Тестируем с помощью пакета doctest.	358
Тестируем с помощью пакета nose.	359
Другие фреймворки для тестирования.	360
Постоянная интеграция.	361
Отлаживаем свой код	361
Отлаживаем с помощью pdb	362
Записываем в журнал сообщения об ошибках	368
Оптимизируем ваш код	371
Измеряем время	371
Алгоритмы и структуры данных.	373
Cython, NumPy и расширения C	374
PyPy	375
Управление исходным кодом	375
Mercurial	375
Git.	376
Клонируйте эту книгу.	378
Как узнать больше.	378
Книги	379
Сайты	379
Группы	380
Конференции	380
Coming Attractions	380

Приложения

Приложение А. Пи-Арт	382
2D-графика	382
Стандартная библиотека	382
PIL и Pillow	383
ImageMagick	386
Графические пользовательские интерфейсы (Graphical User Interface, GUI).	386

Трехмерная графика и анимация	388
Диаграммы, графики и визуализация	391
matplotlib	391
bokeh	392
Игры	393
Аудио и музыка	393
Приложение Б. За работой	395
The Microsoft Office Suite	395
Выполняем бизнес-задачи	397
Обработка бизнес-данных	397
Извлечение, преобразование и загрузка	398
Дополнительные источники информации	401
Python в области финансов	402
Безопасность бизнес-данных	402
Карты	403
Форматы	403
Нарисуем карту	404
Приложения и данные	407
Приложение В. Py в науке	408
Математика и статистика в стандартной библиотеке	408
Математические функции	408
Работа с комплексными числами	410
Рассчитываем точное значение чисел с плавающей точкой с помощью decimal	411
Выполняем вычисления для рациональных чисел с помощью модуля fractions	412
Используем Packed Sequences с помощью array	412
Обработка простой статистики с помощью модуля statistics	413
Перемножение матриц	413
Python для науки	413
NumPy	414
Создание массива с помощью функции array()	414
Создание массива с помощью функции arange()	415

Создание массива с помощью функций <code>zeros()</code> , <code>ones()</code> и <code>random()</code> . . .	416
Изменяем форму массива с помощью метода <code>reshape()</code>	417
Получаем элемент с помощью конструкции <code>[]</code>	418
Математика массивов	419
Линейная алгебра	420
Библиотека <code>SciPy</code>	421
Библиотека <code>SciKit</code>	421
Библиотека <code>IPython</code>	421
Лучший интерпретатор	422
Блокноты <code>IPython</code>	423
<code>Pandas</code>	426
<code>Python</code> и научные области	427
Приложение Г. Установка Python 3	428
Установка стандартной версии <code>Python</code>	428
Mac OS X	431
Windows	432
Linux или Unix	432
Установка <code>Anaconda</code>	432
Установка и использование <code>pip</code> и <code>virtualenv</code>	435
Установка и использование <code>conda</code>	436
Приложение Д. Ответы к упражнениям	437
Глава 1. <code>Python</code> : с чем его едят	437
Глава 2. Ингредиенты <code>Python</code> : числа, строки и переменные	438
Глава 3. Наполнение <code>Python</code> : списки, кортежи, словари и множества . . .	438
Глава 4. Корочка <code>Python</code> : структуры кода	442
Глава 5. <code>Py Boxes</code> : модули, пакеты и программы.	445
Глава 6. Ой-ой-ой: объекты и классы	447
Глава 7. Работаем с данными профессионально	451
Глава 8. Данные должны куда-то попадать	458
Глава 9. Распутываем Всемирную паутину	462
Глава 10. Системы	463
Глава 11. Конкуренция и сети	465

Приложение Е. Вспомогательные таблицы	473
Приоритет операторов	473
Строковые методы	474
Изменение регистра	474
Поиск	474
Изменение	474
Форматирование	475
Тип строки	475
Атрибуты модуля <code>string</code>	476