

Содержание

Предисловие

Об авторе

Часть I. Общие прикладные темы 33

Глава 1. Регулярные выражения 35

1.1. Общее назначение 36

1.1.1. Первое знакомство с регулярными выражениями 38

1.2. Специальные знаки и символы 39

1.2.1. Сопоставление нескольких шаблонов регулярных выражений с помощью оператора чередования (|) 40

1.2.2. Сопоставление с любым отдельным символом (.) 41

1.2.3. Сопоставление с началом или концом строки или с границей слова (^, \$, \b, \B) 41

1.2.4. Создание классов символов ([]) 42

1.2.5. Формирование диапазонов (-) и отрицаний диапазонов (^) 43

1.2.6. Использование операторов замыкания (*, +, ?, {}) для представления нескольких вхождений/повторений 44

1.2.7. Специальные символы, обозначающие наборы символов 45

1.2.8. Обозначение групп с применением круглых скобок (()) 46

1.2.9. Расширенный синтаксис регулярных выражений 47

1.3. Регулярные выражения и язык Python 48

1.3.1. Модуль re. Основные функции и методы 48

1.3.2. Компиляция регулярных выражений с применением функции compile() 50

1.3.3. Объекты сопоставления и методы group() и groups() 51

1.3.4. Согласование со строками с применением функции match() 51

1.3.5. Поиск шаблона в строке с помощью функции search() (поиск вместо сопоставления) 53

1.3.6. Сопоставление с несколькими строками (l) 54

1.3.7. Сопоставление с любым отдельным символом (.) 54

1.3.8. Создание классов символов ([]) 55

1.3.9. Повторение, специальные символы и группирование 55

1.3.10. Сопоставление с началом и концом строк и с границами слов 58

1.3.11. Поиск каждого вхождения с помощью функций findall() и finditer() 58

1.3.12. Поиск и замена с помощью функций sub() и subn() 60

1.3.13. Разбиение (по шаблону разграничения) с помощью метода split() 61

1.3.14. Расширенный синтаксис (?...) 62

1.3.15. Разное 65

1.4. Некоторые примеры регулярных выражений 66

1.5. Более сложный пример регулярного выражения 72

1.5.1. Сопоставление со строкой 74

1.5.2. Сравнение поиска и сопоставления с учетом жадных выражений 76

1.6. Упражнения 79

Глава 2. Сетевое программирование 83

2.1. Введение 84

2.2. Что такое архитектура “клиент–сервер” 84

2.2.1. Аппаратная архитектура “клиент–сервер” 85

2.2.2. Программная архитектура “клиент–сервер” 85

2.2.3. Кассир банка как пример сервера 86

2.2.4. Сетевое программирование по принципу “клиент–сервер” 87

2.3. Сокеты: конечные точки связи	88
2.3.1. Общее определение понятия сокета	88
2.3.2. Адреса сокетов: пара “хост–порт”	89
2.3.3. Сокеты с установлением и без установления соединения	90
2.4. Сетевое программирование на языке Python	91
2.4.1. Функция модуля <code>socket()</code>	91
2.4.2. Методы объекта сокета (встроенные)	92
2.4.3. Создание сервера TCP	93
2.4.4. Создание клиента TCP	97
2.4.5. Эксплуатация сервера и клиентов TCP	101
2.4.6. Создание сервера UDP	102
2.4.7. Создание клиента UDP	103
2.4.8. Эксплуатация сервера и клиентов UDP	105
2.4.9. Атрибуты модуля <code>socket</code>	105
2.5. *Модуль <code>SocketServer</code>	107
2.5.1. Создание сервера TCP с применением модуля <code>SocketServer</code>	108
2.5.2. Создание клиента TCP на основе модуля <code>SocketServer</code>	110
2.5.3. Эксплуатация сервера и клиентов TCP	111
2.6. *Введение в концепцию <code>Twisted</code>	111
2.6.1. Создание сервера TCP на основе классов <code>reactor</code> инфраструктуры <code>Twisted</code>	112
2.6.2. Создание клиента TCP на основе классов <code>reactor</code> инфраструктуры <code>Twisted</code>	113
2.6.3. Эксплуатация сервера и клиентов TCP	115
2.7. Связанные модули	115
2.8. Упражнения	117
Глава 3. Программирование интернет-клиентов	121
3.1. Что такое интернет-клиенты	122
3.2. Передача файлов	123
3.2.1. Протоколы Интернета для передачи файлов	123
3.2.2. Протокол передачи файлов	123
3.2.3. Язык Python и протокол FTP	125
3.2.4. Методы класса <code>ftplib.FTP</code>	126
3.2.5. Пример программы для работы с протоколом FTP в интерактивном режиме	127
3.2.6. Пример клиентской программы FTP	127
3.2.7. Другие особенности протокола FTP	130
3.3. Сетевые новости	131
3.3.1. Usenet и группы новостей	131
3.3.2. Протокол передачи сетевых новостей	132
3.3.3. Язык Python и протокол NNTP	132
3.3.4. Методы класса <code>ntplib.NNTP</code>	134
3.3.5. Пример использования протокола NNTP в интерактивном режиме	135
3.3.6. Пример клиентской программы NNTP	135
3.3.7. Дополнительные сведения о протоколе NNTP	141
3.4. Электронная почта	141
3.4.1. Компоненты и протоколы почтовой системы	141
3.4.2. Отправка электронной почты	142
3.4.3. Язык Python и протокол SMTP	144
3.4.4. Методы класса <code>smtplib.SMTP</code>	145
3.4.5. Пример сеанса интерактивного взаимодействия с использованием протокола SMTP	146
3.4.6. Дополнительные сведения о протоколе SMTP	147
3.4.7. Получение электронной почты	147
3.4.8. Протоколы POP и IMAP	147
3.4.9. Язык Python и протокол POP3	149
3.4.10. Интерактивный пример работы по протоколу POP3	149
3.4.11. Методы класса <code>poplib.POP3</code>	150
3.4.12. Пример применения протоколов SMTP и POP3	151
3.4.13. Язык Python и протокол IMAP4	153
3.4.14. Интерактивный пример применения протокола IMAP4	154

3.4.15. Общие методы класса <code>imaplib.IMAP4</code>	154
3.4.16. Практический пример	156
3.5. Связанные модули	171
3.5.1. Электронная почта	171
3.5.2. Другие клиентские средства поддержки протоколов Интернета	172
3.6. Упражнения	172
Глава 4. Многопоточное программирование	181
4.1. Введение/общее назначение	182
4.2. Потоки и процессы	184
4.2.1. Общее определение понятия процесса	184
4.2.2. Общее определение понятия потока	184
4.3. Поддержка потоков в языке Python	185
4.3.1. Глобальная блокировка интерпретатора	185
4.3.2. Выход из потока	186
4.3.3. Доступ к потокам из программы Python	186
4.3.4. Организация программы без применения потоков	187
4.3.5. Многопоточные модули Python	188
4.4. Модуль <code>thread</code>	189
4.5. Модуль <code>threading</code>	194
4.5.1. Класс <code>Thread</code>	195
4.5.2. Другие функции модуля <code>Threading</code>	202
4.6. Сравнение однопоточного и многопоточного выполнения	203
4.7. Практическое применение многопоточной обработки	205
4.7.1. Пример ранжирования книг	205
4.7.2. Примитивы синхронизации	213
4.7.3. Пример применения блокировки	213
4.7.4. Пример семафора	220
4.8. Проблема “производитель–потребитель” и модуль <code>Queue/queue</code>	225
4.9. Дополнительные сведения об использовании потоков	229
4.9.1. Модуль <code>subprocess</code>	229
4.9.2. Модуль <code>multiprocessing</code>	229
4.9.3. Модуль <code>concurrent.futures</code>	230
4.10. Связанные модули	232
4.11. Упражнения	232
Глава 5. Программирование графического пользовательского интерфейса	235
5.1. Введение	236
5.1.1. Что такое Tcl, Tk и Tkinter	236
5.1.2. Установка и ввод в действие интерфейса Tkinter	237
5.1.3. Архитектура “клиент–сервер” — два компонента	238
5.2. Библиотека Tkinter и программирование на языке Python	238
5.2.1. Модуль Tkinter, обеспечивающий реализацию интерфейса Tk в приложениях	238
5.2.2. Введение в программирование графического пользовательского интерфейса	239
5.2.3. Окно верхнего уровня: <code>Tkinter.Tk()</code>	241
5.2.4. Графические элементы Tk	242
5.3. Примеры Tkinter	243
5.3.1. Графический элемент <code>Label</code>	243
5.3.2. Графический элемент <code>Button</code>	244
5.3.3. Графические элементы <code>Label</code> и <code>Button</code>	245
5.3.4. Графические элементы <code>Label</code> , <code>Button</code> и <code>Scale</code>	246
5.3.5. Более реальный пример	247
5.3.6. Пример использования интерфейса Tkinter в более сложном приложении	251
5.4. Краткий обзор других графических пользовательских интерфейсов	257
5.4.1. Среда Tk Interface eXtensions (Tix)	259
5.4.2. Объекты Python MegaWidgets (PMW)	260
5.4.3. Модули <code>wxWidgets</code> и <code>wxPython</code>	261

5.4.4. Интерфейсы GTK+ и PyGTK	263
5.4.5. Модуль Tile/Ttk	265
5.5. Связанные модули и другие графические пользовательские интерфейсы	268
5.6. Упражнения	271
Глава 6. Программирование баз данных	273
6.1. Введение	274
6.1.1. Система постоянного хранения	274
6.1.2. Основные операции с базами данных и язык SQL	274
6.1.3. Базы данных и язык Python	277
6.2. Спецификация DB-API Python	279
6.2.1. Атрибуты модуля	280
6.2.2. Объекты класса Connection	282
6.2.3. Объекты класса Cursor	283
6.2.4. Объекты и конструкторы типов	285
6.2.5. Реляционные базы данных	286
6.2.6. Базы данных и Python: адаптеры	287
6.2.7. Примеры применения адаптеров баз данных	288
6.2.8. Пример приложения на основе адаптера базы данных	293
6.3. Объектно-реляционные преобразователи	306
6.3.1. Применение объектов вместо запросов SQL	306
6.3.2. Язык Python и объектно-реляционные преобразователи	306
6.3.3. Пример базы данных с описанием должностей сотрудников	308
6.3.4. SQLAlchemy	308
6.4. Нереляционные базы данных	325
6.4.1. Введение в NoSQL	326
6.4.2. База данных MongoDB	326
6.4.3. Адаптер PyMongo: MongoDB и Python	327
6.4.4. Резюме	331
6.5. Справочная информация	332
6.6. Упражнения	334
Глава 7. Программирование приложений для работы с Microsoft Office	339
7.1. Введение	340
7.2. Программирование клиентов COM на языке Python	341
7.2.1. Программирование клиентов COM	341
7.2.2. Вводные сведения	342
7.3. Вступительные примеры	343
7.3.1. Программа Excel	343
7.3.2. Программа Word	346
7.3.3. Программа PowerPoint	347
7.3.4. Программа Outlook	349
7.4. Промежуточные примеры	352
7.4.1. Програма Excel	353
7.4.2. Программа Outlook	355
7.4.3. Программа PowerPoint	362
7.4.4. Резюме	371
7.5. Соответствующие модули/пакеты	372
7.6. Упражнения	372
Глава 8. Создание расширений для языка Python	379
8.1. Введение/обоснование	380
8.1.1. Что такое расширение	380
8.1.2. Причины, по которым может потребоваться создание расширения Python	381
8.1.3. Причины, по которым целесообразно отказаться от создания расширения Python	382
8.2. Модули расширения	383
8.2.1. Создание прикладного кода	383

8.2.2. Оформление кода с применением стандартных шаблонов	385
8.2.3. Компиляция	391
8.2.4. Импорт и проверка	393
8.2.5. Подсчет ссылок	396
8.2.6. Многопоточная организация и GIL	398
8.3. Другие темы	398
8.3.1. Упрощенный генератор оболочек и интерфейса	398
8.3.2. Язык Rurex	399
8.3.3. Язык Cython	399
8.3.4. Язык Pyuso	399
8.3.5. PyRu	400
8.3.6. Внедрение	401
8.4. Упражнения	401
Часть II. Разработка веб-приложений	403
Глава 9. Веб-клиенты и веб-серверы	405
9.1. Введение	406
9.1.1. Навигация по веб-страницам с помощью средств архитектуры клиент–сервер	406
9.1.2. Интернет	407
9.2. Инструменты веб-клиентов Python	410
9.2.1. Унифицированный указатель информационного ресурса	411
9.2.2. Модуль urlparse	412
9.2.3. Модуль/пакет urllib	413
9.2.4. Пример аутентификации с помощью модуля urllib2 при установлении соединения по протоколу HTTP	418
9.2.5. Перенос примера аутентификации HTTP в Python 3	421
9.3. Веб-клиенты	423
9.3.1. Простой поисковый робот, спайдер, бот	423
9.3.2. Синтаксический анализ содержимого веб-страниц	430
9.3.3. Программирование инструментов для просмотра веб-страниц	436
Резюме	440
9.4. Веб-серверы (HTTP)	440
9.4.1. Простые веб-серверы Python	440
9.5. Связанные модули	444
9.6. Упражнения	446
Глава 10. Веб-программирование: интерфейсы CGI и WSGI	451
10.1. Введение	452
10.2. Средства обработки клиентских данных на веб-сервере	452
10.2.1. Введение в интерфейс CGI	452
10.2.2. Приложения CGI	454
10.2.3. Модуль cgi	454
10.2.4. Модуль cgitb	455
10.3. Создание приложений CGI	456
10.3.1. Установка веб-сервера	456
10.3.2. Создание страницы формы	458
10.3.3. Формирование страницы с результатами	459
10.3.4. Формирование страниц с формой и результатами	461
10.3.5. Полностью интерактивные веб-сайты	465
10.4. Использование стандарта кодирования Юникод с интерфейсом CGI	472
10.5. Расширения CGI	473
10.5.1. Передача многокомпонентной формы и передача файла	474
10.5.2. Многозначные поля	474
10.5.3. Cookie-файлы	475
10.5.4. Cookie-файлы и передача файлов	476

10.6. Введение в WSGI	485
10.6.1. Стимулы к дальнейшему развитию (альтернативы технологии CGI)	485
10.6.2. Интеграция сервера	485
10.6.3. Внешние процессы	486
10.6.4. Основные сведения о стандарте WSGI	487
10.6.5. Серверы WSGI	488
10.6.6. Эталонный сервер	489
10.6.7. Примеры приложений WSGI	490
10.6.8. Промежуточное программное обеспечение и создание оболочек для приложений WSGI	491
10.6.9. Изменения в интерфейсе WSGI в версии Python 3	492
10.7. Практическая разработка для веб	493
10.8. Связанные модули	494
10.9. Упражнения	495
Глава 11. Веб-платформы: Django	499
11.1. Введение	500
11.2. Веб-платформы	500
11.3. Введение в Django	502
11.3.1. Установка	503
11.4. Проекты и приложения	507
11.4.1. Создание проекта в Django	507
11.4.2. Работа с сервером для разработки	510
11.5. Первое приложение Hello World (блог)	512
11.6. Создание модели для добавления службы базы данных	513
11.6.1. Подготовка базы данных	514
11.6.2. Создание таблиц	517
11.7. Командный интерпретатор для приложений Python	518
11.7.1. Использование командного интерпретатора Python в Django	519
11.7.2. Эксперименты с моделью данных	520
11.8. Приложение администрирования Django	522
11.8.1. Настройка приложения admin	522
11.8.2. Проверка работы с приложением admin	523
11.9. Создание пользовательского интерфейса для блога	531
11.9.1. Создание шаблона	532
11.9.2. Создание шаблона URL	533
11.9.3. Создание функции представления	537
11.10. Усовершенствование вывода	540
11.10.1. Изменение запроса	541
11.11. Работа с данными, введенными пользователем	545
11.11.1. Шаблон. Добавление формы HTML	546
11.11.2. Добавление записи URLconf	546
11.11.3. Представление. Обработка данных, введенных пользователем	547
11.11.4. Проверка возможности передачи данных с одного сайта на другой	548
11.12. Простые и модельные формы	550
11.12.1. Вводные сведения о формах Django	550
11.12.2. Вариант с применением форм модели	551
11.12.3. Использование объекта класса <code>ModelForm</code> для создания формы HTML	552
11.12.4. Обработка данных класса <code>ModelForm</code>	553
11.13. Дополнительные сведения о представлениях	554
11.13.1. Полууниверсальные представления	555
11.14. *Усовершенствования внешнего интерфейса	557
11.15. *Проверка компонентов	558
11.15.1. Описание кода приложения блога	560
11.15.2. Общие сведения о приложении блога	566
11.16. *Промежуточное приложение Django: <code>TweetApprover</code>	567
11.16.1. Создание структуры файла проекта	568

11.16.2. Установка библиотеки Twython	573
11.16.3. Структура URL	575
11.16.4. Модель данных	579
11.16.5. Передача новых твитов для рецензирования	584
11.16.6. Проверка твитов	589
11.17. Ресурсы	598
11.18. Заключение	599
11.19. Упражнения	599
Глава 12. Облачные вычисления: Google App Engine	605
12.1. Введение	606
12.2. Что такое облачные вычисления	606
12.2.1. Уровни службы облачных вычислений	608
12.2.2. Что такое App Engine	610
12.3. “Песочница” и набор SDK App Engine	613
12.3.1. Службы и API	615
12.4. Выбор платформы для системы App Engine	617
12.4.1. Платформы: webapp, затем Django	619
12.5. Поддержка версии Python 2.7	625
12.5.1. Различия общего характера	626
12.5.2. Различия в коде	626
12.6. Сравнение с платформой Django	627
12.6.1. Запуск приложения Hello World	627
12.6.2. Создание приложения Hello World вручную	628
12.7. Преобразование приложения Hello World в простой блог	629
12.7.1. Быстрый просмотр изменений: преобразование простого текста в HTML за 30 секунд	630
12.7.2. Добавление формы	631
12.7.3. Добавление службы Datastore	633
12.7.4. Постепенные усовершенствования	637
12.7.5. Консоль Разработка/SDK	638
12.8. Добавление службы Memcache	644
12.9. Статические файлы	648
12.10. Добавление службы Users	648
12.10.1. Идентификация с помощью службы Google Accounts	649
12.10.2. Объединенная идентификация	650
12.11. Оболочка удаленного API	650
12.11.1. Функция Datastore Admin	651
12.12. Быстрый обзор (с использованием кода на языке Python)	652
12.12.1. Отправка электронной почты	653
12.12.2. Получение электронной почты	654
12.13. Мгновенная отправка сообщений с помощью службы XMPP	656
12.13.1. Прием при мгновенном обмене сообщениями	657
12.14. Обработка изображений	658
12.15. Очереди задач (незапланированные задачи)	658
12.15.1. Создание задач	659
12.16. Профилирование с помощью Appstats	666
12.16.1. Добавление стандартной программы обработки в файл app.yaml	666
12.16.2. Добавление специализированной страницы Admin Console	667
12.16.3. Инициализация этого интерфейса как встроенной функции	667
12.17. Служба URLfetch	667
12.18. Быстрый обзор (без использования кода Python)	668
12.18.1. Служба Cron (планируемые задачи/задания)	668
12.18.2. Запросы разогрева	669
12.18.3. Защита от хакерских атак типа “отказ в обслуживании”	670
12.19. Обеспечение замкнутости поставщика	670

12.20. Ресурсы	671
12.21. Резюме	673
12.22. Упражнения	674
Глава 13. Веб-службы	679
13.1. Введение	680
13.2. Сервер биржевых котировок Yahoo! Finance	680
13.3. Создание микроблогов в сети Twitter	684
13.3.1. Социальные сети	684
13.3.2. Сеть Twitter и язык Python	685
13.3.3. Пример API с более длинной комбинацией	687
13.3.4. Резюме	699
13.3.5. Дополнительные интернет-ресурсы	699
13.4. Упражнения	699
Часть III. Дополнительная и экспериментальная	705
Глава 14. Обработка текста	707
14.1. Значения, разделяемые запятыми	708
14.1.1. Введение в значения, разделяемые запятыми	708
14.1.2. Повторение примера с портфелем акций	710
14.2. Нотация объектов JavaScript	712
14.3. Расширяемый язык разметки гипертекста	716
14.3.1. Введение в язык XML	716
14.3.2. Языки Python и XML	717
14.3.3. Применение формата XML на практике	721
14.3.4. *Клиент-серверные службы с использованием протокола XML-RPC	725
14.4. Литература	729
14.4.1. Дополнительные ресурсы	729
14.5. Модули, связанные с обработкой текстов	730
14.6. Упражнения	731
Глава 15. Разное	735
15.1. Интерпретатор Jython	736
15.1.1. Введение в интерпретатор Jython	736
15.1.2. Пример графического пользовательского интерфейса с использованием библиотеки Swing	737
15.2. Служба Google+	740
15.2.1. Введение в платформу Google+	740
15.2.2. Среда Python и интерфейс API Google+	741
15.2.3. Простой инструмент анализа социальных средств коммуникации	741
15.3. Упражнения	750
Приложение А . Ответы на некоторые упражнения	755
Приложение Б. Справочные таблицы	759
Приложение В. Версия Python 3: эволюция языка программирования	777
В.1. Почему изменяется язык Python	778
В.2. Что изменилось	778
В.2.1. Оператор print становится функцией print()	778
В.2.2. Строки: Юникод по умолчанию	779
В.2.3. Единственный тип класса	780
В.2.4. Обновленный синтаксис для исключений	780
В.2.5. Обновления, касающиеся целых чисел	781
В.2.6. Повсеместное использование итераторов	782

В.3. Инструменты миграции	783
В.3.1. Инструмент 2to3	783
В.3.1. Версия Python 2.6+	783
В.4. Выводы	784
В.5. Список Литературы	784
Приложение Г. Переход к версии Python 3 на основе выпуска Python 2.6+	785
Г.1. Язык Python 3: следующее поколение	785
Г.1.1. Гибридная версия 2.6+ как инструмент перехода	786
Г.2. Целые числа	787
Г.2.1. Единый целочисленный тип	787
Г.2.2. Новые двоичные и модифицированные восьмеричные литералы	787
Г.2.3. Классическое или истинное деление	788
Г.3. Встроенные функции	790
Г.3.1. Оператор print или функция print ()	790
Г.3.2. Функция reduce () перенесена в модуль functools	790
Г.3.3. Прочие обновления	791
Г.4. Объектно-ориентированное программирование: два разных объекта классов	791
Г.5. Строки	792
Г.5.1. Литералы bytes	793
Г.6. Исключения	793
Г.6.1. Обработка исключений (использование оператора as)	793
Г.6.2. Генерация исключений	794
Г.7. Другие инструменты перехода и рекомендации	794
Г.8. Написание кода, совместимого как с версией 2.x, так и с версией 3.x	795
Г.8.1. Операция print или функция print () ?	796
Г.8.2. Импортируйте в решение свой способ	797
Г.8.3. Составление полной картины из отдельных фрагментов	798
Г.9. Резюме	799
Предметный указатель	801