

Оглавление

Благодарности	25
Об авторе	26
Введение	27
От издательства	29
Глава 1. Начало работы. Программа Game Over	30
Изучаем программу Game Over	30
Знакомство с Python.	31
Python прост в использовании.	31
Python — мощный язык.	32
Python — объектно-ориентированный язык.	32
Python — «склеивающий» язык.	32
Python работает на всех платформах	33
Сообщество программистов на Python.	33
Python — бесплатная система с открытым исходным кодом.	33
Установка Python в Windows	33
Установка Python в других операционных системах	34
Знакомство с IDLE	35
Программирование в интерактивном режиме	35
Программирование в сценарном режиме.	38
Вернемся к программе Game Over	40
Использование комментариев.	41
Пустые строки	41
Печать строки.	42
Ожидаем пользователя.	42
Резюме	42

Глава 2. Типы, переменные и основы ввода-вывода.

Программа «Бесполезные факты»	43
Знакомство с программой «Бесполезные факты»	43
Строки и кавычки	44
Знакомство с программой Game Over 2.0.	44
Кавычки внутри строк.	45
Вывод на экран нескольких значений	46
Задание завершающей строки при выводе	47
Создание строк в тройных кавычках	47
Escape-последовательности в строках	48
Знакомство с программой «Воображаемые благодарности».	48
Вставка табуляционного отступа.	49
Вывод обратного слеша	49
Вставка пустой строки	49
Вставка кавычек.	50
Звук системного динамика	50
Сцепление и повторение строк	51
Знакомство с программой «Забавные строки»	51
Сцепление строк.	52
Символ продолжения строки.	52
Повторение строк.	53
Работа с числами.	53
Знакомство с программой «Текстовые задачи»	53
Числовые типы данных.	55
Применение математических операторов	55
Переменные	56
Знакомство с программой «Привет»	56
Создание переменных.	56
Использование переменных	57
Имена переменных	57
Получение пользовательского ввода	58
Знакомство с программой «Персональный привет»	59
Применение функции input().	59
Применение строковых методов.	61
Знакомство с программой «Манипуляции с цитатой».	61
Создание новых строк с помощью строковых методов.	62
Правильно выбранный тип.	63

Знакомство с программой «Рантье» (версия с ошибкой)	64
Обнаружение и устранение логических ошибок.	65
Конвертация значений.	66
Знакомство с программой «Рантье» (версия без ошибки)	66
Преобразование строк в целые числа	67
Составные операторы присвоения	68
Вернемся к программе «Бесполезные факты»	68
Начальные комментарии.	69
Получение пользовательского ввода.	69
Вывод паме на экран в нижнем и верхнем регистре	69
Пятикратный вывод имени	70
Подсчет количества секунд.	70
Вычисление значений moon_weight и sun_weight	70
Ожидание выхода.	70
Резюме	71

Глава 3. Ветвление, циклы с оператором while и псевдокод.

Игра «Отгадай число»	72
Генерирование случайных чисел	72
Знакомство с программой «Кости»	73
Импорт модуля random	74
Применение функции randint()	74
Применение функции randrange()	75
Условные конструкции с if	75
Знакомство с программой «Пароль»	75
Разбираемся, как работает конструкция if.	77
Создание условий.	77
Операторы сравнения.	77
Создание блоков кода с помощью отступов.	78
Создание собственных условных конструкций.	79
Конструкции if с условием else	79
Знакомство с программой «Открыто/Закрыто»	79
Разбираемся в условиях else.	80
Использование выражения elif	80
Знакомство с программой «Компьютерный датчик настроения».	80
Разбираемся в условиях elif	82

Создание циклов с использованием while	84
Знакомство с программой «Симулятор трехлетнего ребенка»	84
Разбираемся в работе цикла while.	85
Инициализация управляющей переменной	85
Проверка значения управляющей переменной	86
Изменение значения управляющей переменной	86
Борьба с бесконечными циклами	86
Знакомство с программой «Проигранное сражение»	87
Трассировка программы	88
Условия, которые могут становиться ложными	88
Значения как условия	89
Знакомство с программой «Метрдотель».	90
Истинные и ложные значения	91
Намеренное создание бесконечных циклов	91
Знакомство с программой «Привередливая считалка».	92
Выход из цикла с помощью команды break	93
Команда continue и возврат к началу цикла.	93
Как пользоваться командами break и continue	93
Составные условия	93
Знакомство с программой «Эксклюзивная сеть»	94
Логический оператор not	96
Логический оператор and	96
Логический оператор or	97
Планирование программ	98
Алгоритмы на псевдокоде.	98
Пошаговая доработка алгоритма.	99
Вернемся к игре «Отгадай число»	100
План программы.	100
Начальный блок комментариев	101
Импорт модуля random.	101
Объяснение правил	101
Установка начальных значений.	101
Цикл отгадывания	102
Поздравления победителю	102
Ожидание выхода.	102
Резюме	102

Глава 4. Циклы с оператором `for`, строки и кортежи.

Игра «Анаграммы»	104
Знакомство с игрой «Анаграммы»	104
Применение циклов <code>for</code>	105
Знакомство с программой «Слово по буквам»	105
Разбираемся в работе цикла <code>for</code>	106
Создание цикла <code>for</code>	107
Счет с помощью цикла <code>for</code>	107
Знакомство с программой «Считалка»	107
Счет по возрастанию	108
Счет по числам, кратным пяти	109
Счет по убыванию	109
Операторы и функции для работы с последовательностями:	
применение к строкам	109
Знакомство с программой «Анализатор текста»	110
Применение функции <code>len()</code>	110
Применение оператора <code>in</code>	111
Индексация строк	111
Знакомство с программой «Случайные буквы»	111
Позиции с положительными номерами	112
Позиции с отрицательными номерами	113
Случайный элемент строки	114
Что такое неизменяемость строк	115
Конструируем новую строку	116
Знакомство с программой «Только согласные»	116
Создание констант	117
Создание новых строк из существующих	118
Срезы строк	119
Знакомство с программой «Резчик пиццы»	119
Значение <code>None</code>	121
Разбираемся в срезах	121
Создание срезов	122
Сокращения в записи срезов	122
Создание кортежей	123
Знакомство с программой «Арсенал героя»	123
Создание пустого кортежа	124
Кортеж как условие	125

Создание непустого кортежа	125
Вывод элементов кортежа на экран	125
Перебор элементов кортежа	126
Использование кортежей	126
Знакомство с программой «Арсенал героя 2.0»	126
Настройка программы	127
Применение функции len() к кортежам	127
Применение оператора in к кортежам	128
Индексация кортежей	128
Срезы кортежей	128
Неизменяемость кортежей	129
Сцепление кортежей	129
Вернемся к игре «Анаграммы»	130
Настройка программы	130
Как сформировать анаграмму	131
Создание пустой строки для анаграммы	131
Настройка цикла	131
Выбор случайной позиции в слове	132
Новая версия jumble	132
Новая версия word	132
Программа приветствует игрока	132
Получение пользовательского ввода	132
Поздравление с правильно отгаданным словом	133
Конец игры	133
Резюме	133
Глава 5. Списки и словари. Игра «Виселица»	134
Знакомство с игрой «Виселица»	134
Использование списков	135
Знакомство с программой «Арсенал героя 3.0»	136
Создание списка	137
Применение функции len() к спискам	137
Применение оператора in к спискам	137
Индексация списков	138
Срезы списков	138
Сцепление списков	138
Изменяемость списков	138

Присвоение нового значения элементу, выбранному по индексу	139
Присвоение новых значений срезу списка	139
Удаление элемента списка	140
Удаление среза списка	140
Применение списочных методов	140
Знакомство с программой «Рекорды»	140
Настройка программы	141
Отображение меню	141
Выход из программы	142
Отображение списка рекордов	142
Добавление рекорда	142
Удаление рекорда	143
Сортировка списка рекордов	143
Обработка ошибочного выбора	144
Ожидаем пользователя	144
Когда использовать кортежи, а когда — списки	144
Вложенные последовательности	145
Знакомство с программой «Рекорды 2.0»	145
Создаем вложенные последовательности	146
Доступ к вложенным элементам	147
Распаковка последовательности	147
Настройка программы	148
Вывод результатов, содержащихся во вложенных кортежах	148
Добавление результата как вложенного кортежа	149
Обработка ошибочного выбора	149
Ожидаем пользователя	149
Распределенные ссылки	149
Использование словарей	152
Знакомство с программой «Переводчик с гикского на русский»	152
Создание словарей	152
Доступ к значениям в словаре	153
Настройка программы	155
Поиск значения	156
Добавление пары «ключ — значение»	156
Замена пары «ключ — значение»	157
Удаление пары «ключ — значение»	157

Обработка ошибочного выбора	158
Особенности словарей	158
Вернемся к игре «Виселица»	159
Настройка программы	159
Создание констант	159
Инициализация переменных	162
Создание основного цикла	162
Получение ответа игрока	163
Проверка наличия буквы в слове	163
Завершение игры	163
Резюме	164
Глава 6. Функции. Игра «Крестики-нолики»	165
Знакомство с игрой «Крестики-нолики»	165
Создание функций	165
Знакомство с программой «Инструкция»	167
Объявление функции	168
Документирование функции	168
Вызов нестандартной функции	169
Что такое абстракция	169
Параметры и возвращаемые значения	169
Знакомство с программой «Принимай — возвращай»	170
Передача данных с помощью параметров	170
Возврат значений функциями	171
Что такое инкапсуляция	172
Функции, которые и принимают и возвращают значения.	172
Что такое повторное использование кода	173
Именованные аргументы и значения параметров по умолчанию.	174
Знакомство с программой «День рождения»	174
Позиционные параметры и позиционные аргументы	175
Позиционные параметры и именованные аргументы	175
Значения параметров по умолчанию.	176
Использование глобальных переменных и констант	177
Что такое области видимости	177
Знакомство с программой «Доступ отовсюду»	179
Чтение глобальной переменной внутри функции.	180
Затенение глобальной переменной внутри функции	180

Изменение глобальной переменной внутри функции	181
Когда использовать глобальные переменные и константы.	181
Вернемся к игре «Крестики-нолики»	182
План программы «Крестики-нолики».	182
Настройка программы.	184
Функция <code>display_instruct()</code>	185
Функция <code>ask_yes_no()</code>	185
Функция <code>ask_number()</code>	185
Функция <code>pieces()</code>	186
Функция <code>new_board()</code>	186
Функция <code>display_board()</code>	186
Функция <code>legal_moves()</code>	186
Функция <code>winner()</code>	187
Функция <code>human_move()</code>	188
Функция <code>computer_move()</code>	188
Функция <code>next_turn()</code>	191
Функция <code>congrat_winner()</code>	191
Функция <code>main()</code>	192
Запуск программы	192
Резюме	192
Глава 7. Файлы и исключения. Игра «Викторина»	194
Знакомство с игрой «Викторина»	194
Чтение текстового файла.	194
Знакомство с программой «Прочитаем»	195
Открытие и закрытие файла	197
Чтение текстового файла	198
Посимвольное чтение строки	199
Чтение всех строк файла в список	200
Перебор строк файла	200
Запись в текстовый файл.	200
Знакомство с программой «Запишем»	201
Запись строк в файл	201
Запись списка строк в файл	202
Хранение структурированных данных в файлах	203
Знакомство с программой «Законсервируем»	203
Консервация данных и запись в файл	204

Чтение и расконсервация данных из файла	205
Полка для хранения консервированных данных	206
Извлечение консервированных данных через интерфейс полки.	207
Обработка исключений	208
Знакомство с программой «Обрабатываем»	208
Применение конструкций try/except	208
Типы исключений	209
Обработка нескольких типов исключений	211
Аргумент исключения	212
Добавление блока else	212
Вернемся к игре «Викторина»	212
Как организованы данные в текстовом файле	213
Функция open_file().	214
Функция next_line().	215
Функция next_block()	215
Функция welcome().	216
Настройка игры	216
Задание вопроса	216
Получение ответа	216
Проверка ответа	217
Переход к следующему вопросу	217
Завершение игры	217
Запуск функции main()	217
Резюме	217

Глава 8. Программные объекты. Программа

«Моя зверюшка»	219
Знакомство с программой «Моя зверюшка»	219
Основы объектно-ориентированного подхода	221
Создание классов, методов и объектов	221
Знакомство с программой «Просто зверюшка»	221
Объявление класса	222
Объявление метода	223
Создание объекта	223
Вызов метода	224
Применение конструкторов	224
Знакомство с программой «Зверюшка с конструктором»	224

Создание конструктора	225
Создание нескольких объектов	225
Применение атрибутов	226
Знакомство с программой «Зверюшка с атрибутом»	226
Инициализация атрибутов	227
Доступ к атрибутам	228
Вывод объекта на экран	228
Применение атрибутов класса и статических методов	229
Знакомство с программой «Классово верная зверюшка»	229
Создание атрибута класса.	231
Доступ к атрибуту класса	231
Создание статического метода	232
Вызов статического метода.	232
Что такое инкапсуляция объектов	232
Применение закрытых атрибутов и методов.	233
Знакомство с программой «Закрытая зверюшка».	233
Создание закрытых атрибутов	234
Доступ к закрытым атрибутам.	234
Создание закрытых методов	235
Доступ к закрытым методам	235
Соблюдаем приватность	236
В каких случаях нужны закрытые атрибуты и методы	237
Управление доступом к атрибутам	237
Знакомство с программой «Зверюшка со свойствами».	237
Создание свойств	238
Доступ к свойствам.	239
Вернемся к программе «Моя зверюшка»	240
Класс Critter	241
Создание зверюшки	242
Создание меню.	243
Запуск программы	243
Резюме	244
Глава 9. Объектно-ориентированное программирование.	
Игра «Блек-джек»	245
Знакомство с игрой «Блек-джек»	245
Отправка и прием сообщений	246

Знакомство с программой «Гибель пришельца»	247
Отправка сообщения	248
Прием сообщения.	249
Сочетание объектов.	249
Знакомство с программой «Карты»	249
Создание класса Card	250
Создание класса Hand.	250
Применение объектов-карт.	251
Сочетание объектов-карт в объекте Hand	251
Создание новых классов с помощью наследования.	252
Расширение класса через наследование.	253
Знакомство с программой «Карты 2.0»	253
Создание базового класса.	253
Наследование от базового класса	255
Расширение производного класса	255
Применение производного класса	256
Переопределение унаследованных методов	258
Знакомство с программой «Карты 3.0»	258
Создание базового класса.	258
Переопределение методов базового класса.	259
Вызов методов базового класса	260
Применение производного класса.	261
Что такое полиморфизм.	262
Создание модулей	262
Знакомство с программой «Простая игра».	263
Пишем модуль	263
Импорт модулей	265
Применение импортированных функций и классов	265
Вернемся к игре «Блек-джек»	266
Модуль cards	266
Продумаем систему классов	267
Напишем псевдокод для основного цикла игры.	268
Импорт модулей cards и games	269
Класс BJ_Card.	269
Класс BJ_Deck.	270
Класс BJ_Hand	270
Класс BJ_Player.	272

Класс VJ_Dealer	272
Класс VJ_Game	273
Функция main().	275
Резюме	276

Глава 10. Разработка графических интерфейсов.

Программа «Сумасшедший сказочник»	277
Знакомство с программой «Сумасшедший сказочник»	277
GUI в подробностях	279
Что такое событийно-ориентированное программирование	280
Базовое окно	281
Знакомство с программой «Простейший GUI».	281
Импорт модуля tkinter	282
Создание базового окна	283
Изменение вида базового окна	283
Запуск событийного цикла базового окна	283
Применение меток	284
Знакомство с программой «Это я, метка»	284
Настройка программы	284
Создание рамки	284
Создание метки	285
Запуск событийного цикла базового окна	285
Применение кнопок	285
Знакомство с программой «Бесполезные кнопки»	286
Настройка программы	286
Создание кнопок	286
Запуск событийного цикла базового окна	287
Создание GUI с помощью класса	287
Знакомство с программой «Бесполезные кнопки — 2».	288
Импорт модуля tkinter	288
Объявление класса Application	288
Объявление метода-конструктора	288
Объявление метода, создающего элементы управления	289
Создание объекта класса Application	289
Связывание элементов управления с обработчиками событий	290
Знакомство с программой «Счетчик щелчков».	290
Настройка программы	290
Связывание обработчика с событием	291

Создание обработчика события.	291
Обертка программы	291
Текстовые поля и области. Менеджер размещения Grid	291
Знакомство с программой «Долгожитель».	292
Настройка программы.	293
Размещение элементов управления с помощью менеджера Grid	293
Создание текстового поля.	294
Создание текстовой области	295
Текстовые элементы: извлечение и вставка данных	295
Обертка программы	296
Применение флажков	296
Знакомство с программой «Киноман»	297
Настройка программы.	297
Ссылка только на родительский объект элемента управления	297
Создание флажков	298
Получение статуса флажка	299
Обертка программы	300
Применение переключателей.	300
Знакомство с программой «Киноман-2».	300
Настройка программы.	301
Создание переключателя	301
Доступ к значениям в переключателе	303
Обертка программы	303
Вернемся к программе «Сумасшедший сказочник»	303
Импорт модуля tkinter.	303
Метод-конструктор класса Application	304
Метод create_widgets() класса Application	304
Метод tell_story() класса Application.	305
Основная часть программы.	306
Резюме	307
Глава 11. Графика. Игра «Паника в пиццерии»	308
Знакомство с игрой «Паника в пиццерии»	308
Знакомство с пакетами pygame и livewires	308
Создание графического окна	310
Знакомство с программой «Новое графическое окно»	310
Импорт модуля games.	311

Инициализация графического экрана	312
Запуск основного цикла	312
Назначение фоновой картинки	313
Знакомство с программой «Фоновая картинка»	313
Загрузка изображения	313
Установка фона	315
Что такое система координат графики	315
Отображение спрайта	315
Знакомство с программой «Спрайт-пицца»	317
Загрузка изображения для спрайта	318
Создание спрайта	318
Добавление спрайта на экран	319
Отображение текста	320
Знакомство с программой «Ничего себе результат!»	320
Импорт модуля color	321
Создание объекта Text	322
Добавление объекта Text на экран	322
Вывод сообщения	323
Знакомство с программой «Победа»	323
Импорт модуля color	324
Создание объекта Message	324
Ширина и высота графического экрана	325
Добавление объекта Message на экран	325
Подвижные спрайты	325
Знакомство с программой «Летающая пицца»	326
Настройка скорости движения спрайта	327
Учет границ экрана	327
Программа «Скачущая пицца»	327
Настройка программы	328
Создание подкласса Sprite	328
Переопределение метода update()	329
Обертка программы	329
Обработка ввода с помощью мыши	330
Знакомство с программой «Подвижная сковорода»	330
Настройка программы	330
Чтение координат указателя	331
Настройка видимости указателя	332

Перенаправление ввода в графическое окно	332
Обертка программы	332
Регистрация столкновений	333
Знакомство с программой «Ускользящая пицца»	333
Настройка программы	333
Регистрация столкновений	335
Обработка столкновений	335
Обертка программы	335
Вернемся к игре «Паника в пиццерии»	336
Настройка программы	336
Класс Pan	336
Класс Pizza	338
Класс Chef	340
Функция main()	341
Резюме	342

Глава 12. Звук, анимация, разработка больших программ.

Игра «Прерванный полет»	343
Знакомство с игрой «Прерванный полет»	343
Чтение с клавиатуры	345
Знакомство с программой «Читаю с клавиатуры»	345
Настройка программы	346
Регистрация нажатий	346
Обертка программы	347
Вращение спрайта	347
Знакомство с программой «Крутящийся спрайт»	347
Применение свойства angle у спрайтов	349
Создание анимации	349
Знакомство с программой «Взрыв»	349
Посмотрим на картинки	350
Настройка программы	350
Создание списка изображений	351
Создание анимированного объекта	352
Работа со звуком и музыкой	352
Знакомство с программой «Звук и музыка»	352
Работа со звуком	353
Работа с музыкой	355
Обертка программы	357

Разработка игры «Прерванный полет»	357
Функциональность игры	357
Классы игры	358
Медиаресурсы	358
Создание астероидов	358
Программа «Прерванный полет — 1»	359
Настройка программы	359
Класс Asteroid	360
Функция main()	361
Вращение корабля	361
Программа «Прерванный полет — 2»	361
Класс Ship	362
Инстанцирование класса Ship	362
Движение корабля	363
Программа «Прерванный полет — 3»	363
Импорт модуля math	364
Добавление переменной и константы в класс Ship	364
Изменение метода update() объекта Ship	364
Стрельба ракетами	365
Программа «Прерванный полет — 4»	366
Изменение метода update() объекта Ship	366
Класс Missile	367
Управление плотностью огня	369
Программа «Прерванный полет — 5»	369
Добавление константы в класс Ship	370
Создание метода-конструктора в классе Ship	370
Изменение метода update() объекта Ship	370
Обработка столкновений	371
Программа «Прерванный полет — 6»	371
Изменение метода update() объекта Missile	371
Добавление метода die() объекту Missile	371
Изменение метода update() объекта Ship	372
Добавление метода die() объекту Ship	373
Добавление константы в класс Asteroid	373
Добавление метода die() объекту Asteroid	373
Добавление взрывов	373
Программа «Прерванный полет — 7»	374
Класс Wrapper	374

Класс Collider	375
Изменение класса Asteroid	376
Изменение класса Ship	376
Изменение класса Missile.	377
Класс Explosion	377
Уровни, ведение счета, музыкальная тема	378
Программа «Прерванный полет — 8»	378
Импорт модуля color	378
Класс Game	379
Добавление переменной и константы в класс Asteroid	383
Изменение метода-конструктора в классе Asteroid.	383
Изменение метода die() объекта Asteroid.	383
Добавление константы в класс Ship	384
Изменение метода-конструктора в классе Ship	384
Изменение метода update() объекта Ship.	384
Добавление метода die() объекту Ship.	385
Функция main().	385
Резюме	385
Приложение А. Сайт-помощник	387
Приложение В. Справка по пакету livewires.	388
Пакет livewires.	388
Классы модуля games	388
Класс Screen.	389
Класс Sprite	390
Класс Text.	391
Класс Message	392
Класс Animation	393
Класс Mouse	393
Класс Keyboard	394
Класс Music.	394
Функции модуля games	395
Константы модуля games.	396
Константы модуля color	399
Алфавитный указатель.	400

1 Начало работы. Программа Game Over

Программировать — значит, по сути, заставлять компьютер что-то делать. Таково далекое от технических деталей, но очень точное определение. Изучая Python, вы научитесь писать программы: от простых игр и небольших утилит до бизнес-приложений с богатым графическим интерфейсом (GUI). Эти программы будут плодами лично вашего труда, и работать они будут именно так, как вы им предпишете.

В программировании есть доля науки, доля искусства и огромная примесь приключенческого романа. С этой главы начнутся ваши приключения в мире Python. Здесь вам предстоит узнать:

- что представляет собой Python и чем он интересен;
- как установить Python на ваш компьютер;
- как выводить текст на экран;
- что такое комментарии в коде и как ими пользоваться;
- как с помощью интегрированной среды разработки Python писать, редактировать, запускать и сохранять программы.

Изучаем программу Game Over

Программа Game Over, которой посвящена эта глава, выводит на экран два слова, печально известных всем любителям компьютерных игр: Game Over. На рис. 1.1 показано, как работает программа.

На рис. 1.1 показана так называемая *консоль* — окно, способное отображать только текст. Консольные приложения гораздо менее симпатичны, чем программы с графическим интерфейсом (GUI), но они проще в написании, что делает их удобными для начинающего программиста.

Программа Game Over исключительно проста; в сущности, это одна из самых простых программ, которые вообще можно написать на языке Python. Вот почему я и посвятил ей эту главу. В ходе разработки столь скромной программы вы без помех выполните всю подготовительную работу, которая нужна, чтобы начать программировать на Python. В частности, понадобится установить интерпретатор языка в вашей операционной системе. Вам также предстоит проработать весь цикл написания, сохранения и запуска программы. Окунувшись в первый раз в эту рутину, вы подготовитесь к созданию более крупных и полезных приложений.

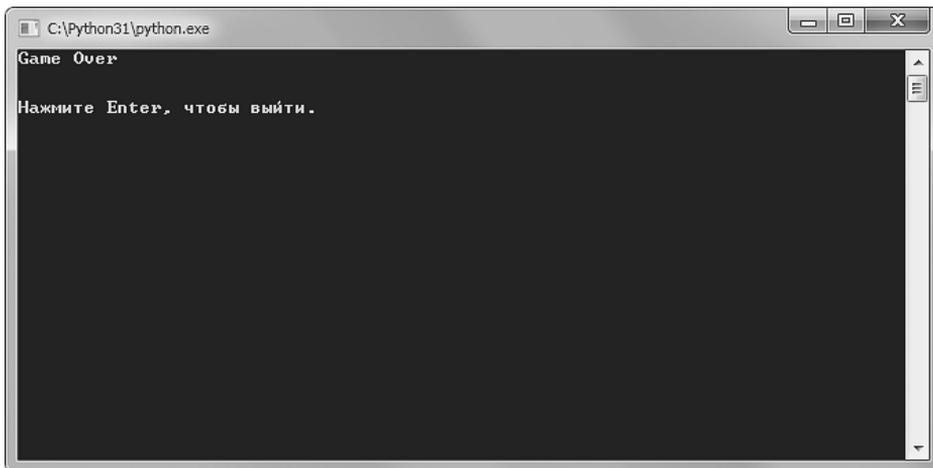


Рис. 1.1. Каждому, кто играл на компьютере, эти слова знакомы

НА САМОМ ДЕЛЕ

Программа Game Over — разновидность традиционной программы Hello World, выводящей на экран слова Hello World («Привет, мир»). Обычно Hello World — самая первая программа, которую пишет начинающий программист при знакомстве с новым для него языком. Эта практика столь широко распространена, что Hello World — термин, хорошо понятный каждому программисту.

Знакомство с Python

Python — мощный и простой в использовании язык программирования, разработанный Гвидо ван Россумом (Guido van Rossum). Первый релиз системы вышел в 1991 году. На Python можно быстро написать небольшой проект, а вообще он применим к проектам любого масштаба, в том числе коммерческим приложениям и программам, нацеленным на ответственные задачи. При знакомстве с документацией Python вас может удивить обилие упоминаний про число 42, спам и яйца. Все это осталось в память об английском коллективе комических актеров «Монти Пайтон» (Monty Python), в честь которого и был назван язык. Хотя Гвидо ван Россум произвел название Python от названия комик-труппы, официальным символом языка стала змейка питон (других вариантов, пожалуй, и не оставалось, ведь значок на Рабочем столе слишком мал, чтобы уместить на нем лица шести английских юмористов).

Существует много других языков программирования. Чем же так привлекателен Python? Судите сами.

Python прост в использовании

Базовая цель любого языка программирования — построить «мостик» между мозгом программиста и компьютером. Те популярные языки, о которых вы, вероятно,

слышали, — Visual Basic, C#, Java и др. — принято называть *языками высокого уровня*. Этот термин предполагает, что они ближе к человеческому языку, чем к машинному. Так оно и есть. Но Python с его простыми и ясными правилами еще приближается к английскому языку. Создание программ на Python настолько незамысловатый процесс, что о нем иногда говорят как о «программировании со скоростью мысли». С точки зрения профессионального программиста, легкость Python — залог высокой производительности труда: программы на Python короткие и требуют меньше времени на разработку, чем программы на многих других популярных языках.

Python — мощный язык

Python обладает всеми возможностями, которых следует ожидать от современного языка программирования. Читая эту книгу, вы научитесь пользоваться разнообразными структурами данных и писать программы с GUI и файлами состояния.

Благодаря своей мощности Python привлекает разработчиков со всего мира. Им пользуются крупнейшие компании: Google, IBM, Industrial Light + Magic, Microsoft, NASA, Red Hat, Verizon, Xerox и Yahoo!. Профессиональные разработчики игр также применяют Python. Electronic Arts, 2K Games и Disney Interactive Media Group — все эти компании публикуют игры с кодом на Python.

Python — объектно-ориентированный язык

Объектно-ориентированное программирование (ООП) — современный подход к решению задач с помощью вычислительных машин. В рамках ООП собственная информация программы и команды, которые она передает компьютеру, записываются интуитивно понятным образом. Это, конечно, не единственный способ разработки программ, но в больших проектах, как правило, предпочтительный.

C#, Java и Python — объектно-ориентированные языки. Но у Python есть преимущество перед первыми двумя. В C# и Java ООП-подход проводится неукоснительно. Это делает короткие программы избыточно сложными; прежде чем начинающий программист сумеет сделать что-либо толковое, его нужно долго учить языку. В Python заложена другая модель: ООП-приемами пользоваться не обязательно. Вся их мощь по-прежнему находится в вашем распоряжении, но только вам дано решать, когда именно воспользоваться этой мощью. Ваша простенькая программа не требует ООП? Все в порядке. Ваш большой проект с целой командой программистов жизненно нуждается в ООП? Тоже не проблема. Python совмещает в себе функциональность и гибкость.

Python — «склеивающий» язык

Python легко интегрировать с другими языками, например C, C++ или Java. Таким образом, программист на Python может пользоваться разработками, уже имеющимися на других языках, и обращать в свою пользу сильные стороны этих языков (скажем, быстродействие C/C++), не поступаясь простотой разработки — отличительной чертой Python.

Python работает на всех платформах

Python-программу можно запустить на любой машине: от миниатюрного Palm до суперкомпьютера Cray. Если для суперкомпьютера вы недостаточно богаты, то пользоваться Python сможете на ПК с операционной системой Windows, Macintosh, Linux — и это лишь часть списка.

Программы на Python *независимы от платформы*, то есть неважно, какой операционной системой пользовался разработчик программы: код может быть исполнен на любом компьютере с установленным интерпретатором Python. Если, например, вы, работая на PC-совместимом компьютере, написали программу, то ничто не мешает вам переслать копии этой программы другу-линуксу и бабушке — обладательнице машины Macintosh. Если на компьютерах друга и бабушки установлен Python, программа будет функционировать.

Сообщество программистов на Python

Почти каждому языку программирования посвящена хотя бы новостная группа. Среди ресурсов о Python есть особая рассылка Python Tutor, в рамках которой новички могут в свободной форме задавать вопросы о языке. Адрес этой рассылки: <http://mail.python.org/mailman/listinfo/tutor>. Несмотря на то что в названии есть слово Tutor («наставник»), на вопросы может отвечать кто угодно: и эксперт, и начинающий.

Существуют и другие сообщества, посвященные разным вопросам использования Python. Их общая черта в том, что все они чрезвычайно открытые, дружелюбные по отношению к новым участникам. Неудивительно, раз уж язык так прост для освоения с нуля.

Python — бесплатная система с открытым исходным кодом

Интерпретатор Python бесплатен. Чтобы установить его и пользоваться им, не надо платить ни копейки. Лицензия, под которой доступен Python, разрешает и многие другие интересные вещи. Можно делать копии Python, в том числе модифицированные. Если угодно, можно даже продать Python (но не спешите бросать чтение — вы еще успеете получить свою прибыль!). Следование концепции открытого исходного кода — одна из многих особенностей, делающих Python столь популярным и успешным.

Установка Python в Windows

Прежде чем писать свою первую программу на Python, вы должны установить систему программирования на своем компьютере. Не беспокойтесь: вся процедура установки Python на компьютере с операционной системой Windows сейчас будет изложена.

Для установки интерпретатора Python в Windows сделайте следующее.

1. Скачайте установочный файл Python Windows с сайта-помощника (www.courseptr.com/downloads). Файл находится в подкаталоге Python каталога Software и носит название `python-3.1.msi`.
2. Запустите `python-3.1.msi` — установочный файл Python для Windows. Процесс установки показан на рис. 1.2.
3. Подтвердите конфигурацию установки по умолчанию. Как только вы это сделаете, Python 3.1 появится на вашем компьютере.

ПОДСКАЗКА

www.courseptr.com/downloads — сайт-помощник, на котором размещены сопроводительные материалы к этой книге. Там, в частности, можно найти код всех законченных программ, о которых здесь пойдет речь, а также все необходимые установочные файлы и документацию. Более подробное описание см. в приложении А.



Рис. 1.2. Уже совсем скоро Python поселится и на вашем компьютере

Установка Python в других операционных системах

Python реализован буквально для нескольких десятков операционных систем. Поэтому, если вы пользуетесь не Windows, вам достаточно будет посетить официальный сайт Python <http://www.python.org> и скачать оттуда последнюю версию интерпретатора для вашей платформы. Вид главной страницы сайта Python показан на рис. 1.3.



Рис. 1.3. Посетив официальный сайт Python, вы сможете скачать самую свежую версию системы и узнать много нового о языке

ЛОВУШКА

Может быть так, что на вашем компьютере уже установлен интерпретатор Python. Помните, что для корректной работы программ, представленных в этой книге, нужен Python версии 3.

Знакомство с IDLE

В комплект поставки Python входит интегрированная среда разработки под названием IDLE. *Средой разработки* называется совокупность инструментов, которые упрощают создание программ. Это в каком-то роде текстовый процессор для ваших программ. Но среда разработки позволяет не только писать, сохранять и редактировать программы. У IDLE есть два режима работы: интерактивный и сценарный.

Программирование в интерактивном режиме

Вот и пришло для вас время узнать, каково оно на вкус — настоящее программирование на Python. Для этого запустим Python в интерактивном режиме, в котором пользователь сообщает системе, что надо сделать, а та немедленно отвечает.

Ваша первая программа

Запустить интерактивную сессию можно так. В меню Пуск выберите Программы ▶ Python 3.1 ▶ IDLE (Python GUI). На экране появится окно, очень похожее на то, которое показано на рис. 1.4.

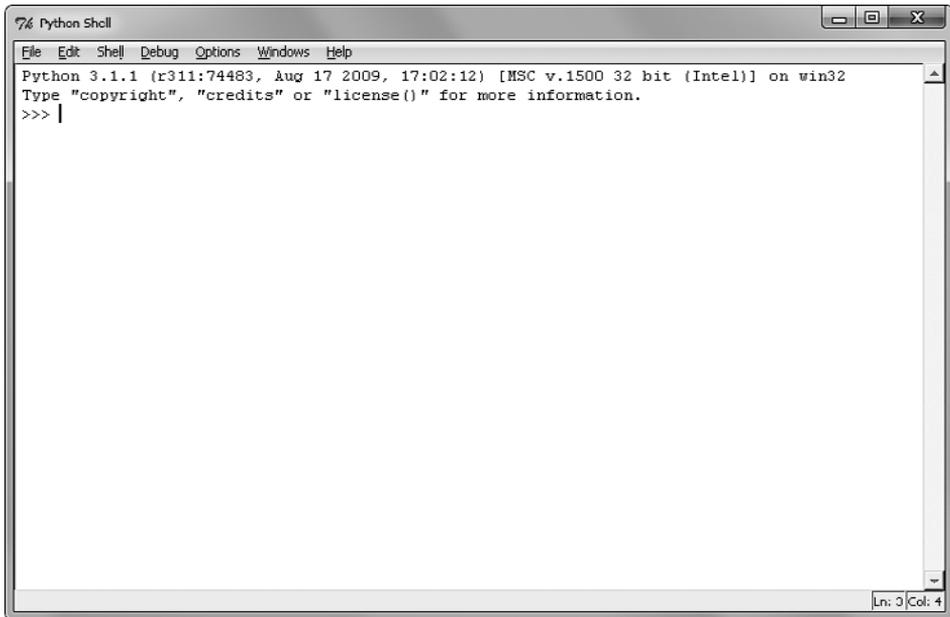


Рис. 1.4. Интерактивная сессия работы с Python. Система ожидает ваших команд

Это так называемый терминал Python (Python Shell). В командной строке (>>>) введите `print("Game Over")` и затем нажмите клавишу `Enter`. В ответ интерпретатор выведет на экран текст:

```
Game Over
```

Итак, вы только что написали свою первую программу на Python! Теперь можно считать вас настоящим программистом. Правда, надо еще кое-чему научиться, но это касается нас всех.

Использование функции `print`

Присмотритесь к строке, которую вы ввели: `print("Game Over")`. Она очень незатейлива, как видите. Не зная ничего о программировании (и владея английским хотя бы на базовом уровне), можно догадаться, что этот код значит. Таков по своей сути весь Python. Его визитная карточка — краткость и ясность, которые вам еще предстоит оценить по достоинству в ходе изучения кое-чего посложнее.

Функция `print()` выводит на экран текст, который пользователь, заключив в кавычки, помещает внутрь скобок. Если ничем не заполнить скобки, будет напечатана пустая строка.

ЛОВУШКА

Python чувствителен к регистру. Существует договоренность, в силу которой названия функций пишутся строчными буквами. Поэтому запись `print(«Game Over»)` сработает, а `Print(«Game Over»)` или `PRINT(«Game Over»)` — нет.
