

Содержание

	<i>Предисловие</i>	13
	<i>Благодарности</i>	17
	<i>Об этой книге</i>	19
	<i>От издательства</i>	23
1	Приступая к работе	24
	Установка Python	24
	Взаимодействие с Python посредством среды разработки IDLE.....	25
	Введи инструкции, пожалуйста	26
	Взаимодействие с Python.....	29
	Время программировать.....	31
	Запуск первой программы.....	32
	Если что-то пойдет не так.....	33
	Наша вторая программа	35
2	Память и переменные	38
	Ввод, обработка, вывод	38
	Имена	40
	Что в имени моем?	45
	Числа и строки	46
	Насколько они «переменные»?.....	47
	Новый я	48
3	Основы математики	52
	Четыре базовые операции	53
	Операторы	53
	Порядок операций.....	54
	Целочисленное деление: частное и остаток.....	56
	Потенцирование – возведение в степень	57
	Приращение и уменьшение.....	59
	Действительно большой и действительно маленький	59
4	Типы данных	64
	Изменение типа.....	64
	Как получить больше информации: <code>type()</code>	68

6	Содержание	
	Ошибки при изменении типа.....	68
5	Ввод	70
	Функция <code>input()</code>	71
	Использование функции <code>input()</code> в одной строке.....	71
	Ввод чисел.....	73
	Ввод из Всемирной паутины.....	74
6	Графические пользовательские интерфейсы	77
	Что такое GUI?.....	77
	Наш первый GUI.....	78
	Ввод с помощью GUI.....	79
	Выбери на свой вкус.....	80
	И снова игра по угадыванию чисел.....	83
	Другие части GUI.....	84
7	Вычисления	86
	Тестирование.....	86
	Выделение отступами.....	88
	У меня двоится в глазах?.....	89
	Другие виды тестов.....	90
	Что случается, если результат проверки оказался ложным?.....	91
	Проверка более чем одного условия.....	93
	Использование ключевого слова <code>and</code>	93
	Использование ключевого слова <code>or</code>	94
	Использование ключевого слова <code>not</code>	95
8	Циклы	99
	Циклы со счетчиком.....	100
	Использование цикла со счетчиком.....	102
	Сокращение – <code>range()</code>	103
	Вопрос стиля – имена переменных цикла.....	104
	Счетчик по шагам.....	107
	Счетчик без чисел.....	108
	Раз мы коснулись этой темы... ..	109
	Выход из цикла – <code>break</code> и <code>continue</code>	110
9	Комментарии	114
	Добавление комментариев.....	114
	Однострочные комментарии.....	115
	Комментарии в конце строки.....	115
	Многострочные комментарии.....	116
	Строки в тройных кавычках.....	116
	Стиль комментирования.....	117
	Комментарии в этой книге.....	117
	Комментирование кода.....	117

10	Время поиграть	119
	Игра «Лыжник»	119
11	Вложенные и переменные циклы	123
	Вложенные циклы	123
	Переменные циклы	125
	Вложенные переменные циклы	126
	Еще более глубоко вложенные циклы с переменной	128
	Использование вложенных циклов.....	129
	Подсчет калорий.....	132
12	Собираем все воедино: списки и словари	137
	Что такое список?	137
	Создание списка	138
	Добавление элементов в список.....	138
	Получение элементов из списка.....	140
	«Нарезка» списка	141
	Изменение элементов	143
	Другие способы добавления элементов в список	144
	Удаление элементов из списка	145
	Поиск по списку.....	147
	Сортировка списков	149
	Изменяемый и неизменяемый.....	153
	Списки списков: таблицы данных.....	153
	Словари	157
13	Функции	164
	Функции – строительные кирпичики	164
	Передача аргументов функции	168
	Функции, возвращающие значение.....	172
	Область видимости переменной.....	174
	Небольшой совет касательно присвоения имен переменным	177
14	Объекты	180
	Объекты в реальном мире	180
	Объекты в Python.....	181
	Объект = атрибуты + методы	182
	Создание объектов	183
	Пример класса – хот-дог	188
	Соккрытие данных	192
	Полиморфизм и наследование	193
	Планирование	196
15	Модули	198
	Что такое модуль?.....	198

Зачем использовать модули	198
Как мы создаем модули?.....	199
Как мы используем модули?.....	200
Пространство имен	201
Стандартные модули.....	204
16 Графика.....	209
Небольшая помощь – Pygame	209
Окно Pygame	210
Рисование в окне	211
Отдельные пиксели	220
Изображения.....	225
Давай двигаться!.....	227
Анимация	228
Плавная анимация	229
Мяч отталкивается	230
Мяч оборачивается.....	232
17 Спрайты и обнаружение столкновений.....	236
Спрайты	236
Бах! Обнаружение столкновений	242
Отсчет времени	246
18 Новый вид ввода – события.....	251
События.....	251
События клавиатуры	253
События мыши	257
События таймера.....	259
Еще одна игра – PyPong.....	261
19 Звуковое сопровождение.....	273
Модуль <code>pygame.mixer</code>	273
Создание и воспроизведение звуков	274
Воспроизведение звуков.....	274
Управление уровнем громкости.....	277
Воспроизведение фоновой музыки	278
Повтор музыки	279
Добавление звукового сопровождения в PyPong	280
Добавление музыки в PyPong.....	284
20 Продолжение работы над графическими интерфейсами	289
Работа с PyQt	289
Qt Designer	290
Сохранение проекта GUI.....	293
Как назначить действие элементу GUI	295
Возвращение обработчиков событий	297

Полезные GUI	298
TempGUI	299
Создание нового GUI	300
Исправление бага	304
Содержимое меню	305
21 Форматирование вывода	312
Переход строки	313
Горизонтальное выделение пробелами – табуляция	315
Вставка переменных в строки	317
Форматирование чисел	318
Новый способ форматирования	323
Форматирование строк	325
22 Ввод и вывод файла	333
Что такое файл?	334
Имена файлов	334
Расположение файлов	335
Открытие файла	338
Чтение файла	339
Текстовые и двоичные файлы	342
Запись в файл	343
Сохранение данных в файл	347
И снова игра – «Виселица»	349
23 Аспект случайности	357
Что такое случайный порядок?	358
Бросаем кости	358
Создание колоды карт	363
Сумасшедшие восьмерки	368
24 Компьютерное моделирование	381
Моделирование реального мира	381
Игра «Луноход»	382
Следим за временем	387
Временные объекты	388
Игра «Тамагочи»	393
25 А теперь поподробнее о «Лыжнике»	405
«Лыжник»	405
Препятствия	409
26 Создание сетевых соединений с помощью сокетов	419
В чем разница между текстом и байтами?	420
Серверы	422

10 Содержание

Получение данных от клиента.....	426
Создание чат-сервера.....	426
27 Что дальше?	440
Юным программистам.....	440
Python	441
Разработка игр и Pygame	441
Создание игр на других языках (не на Python)	441
Да будет BASIC	442
Создание веб-сайтов	442
Создание мобильных приложений	442
Оглянись вокруг.....	442
A Правила присвоения имен переменным	444
Б Разница между Python 3 и Python 2	446
В Ответы на вопросы	450
<i>Предметный указатель</i>	483

Отзывы о первом издании книги

Отличная книга, подойдет как для маленьких, так и для больших деток.

– Гордон Колкхаун,
компьютерный консультант в компании Avalon Consulting Services

Python для подрастающих.

– Доктор Джон Грейсон, автор программ на Python и Tkinter

Книга, с которой интересно учиться!

– Доктор Андре Роберж, президент Университета Святой Анны

Авторы создали прекрасную книгу по программированию для увлекательного обучения.

– Брайан Вайнгартен, архитектор программного обеспечения

Я очень рекомендую эту книгу!

– Хорст Йенс, преподаватель языка Python
и автор книги «*Programming While Playing*»

Язык Python прекрасно подходит для новичков. Приятно видеть ориентированную на ребенка книгу о Python!

– Джеффри Элкнер, педагог

Если вы можете научить своего ребенка чему-то одному, научите его золотому правилу нравственности. Если вы можете научить ребенка двум вещам, научите его золотому правилу и компьютерному программированию. Для обучения программированию этой книги будет более чем достаточно.

– Джош Кронмайер,
старший консультант по программному обеспечению
в компании Thoughtworks

Мне очень понравилось «общение» с Картером в этой книге... А мои ученики будут в восторге от виртуального питомца! Он напоминает мне Тамагочи, который был у меня много лет назад.

– Кари Дж. Стеллпфлаг,
преподаватель государственной школы Рочестера, Миннесота

Компьютерное программирование – это мощный инструмент для обучения детей... Дети, которые занимаются программированием, переносят полученные навыки и в другие сферы.

– Николас Негропонте, участник проекта «One Laptop Per Child»

Отзывы о втором издании книги

Самое простое обучение программированию!

– Шон Стебнер, инженер сетевого оборудования в компании Intel Corp

Прочитав эту книгу, вы поймете, что программировать так же легко, как пожарить яичницу.

– Элизабет Гордон, ученица 10-го класса средней школы Игл Харбор

Отличное введение в мир Python для всех. Эта книга очень увлекательна!

– Мейсон Дженкинс, ученик 7-го класса Академии Майрона Б. Томпсона

Книга для людей от 8 до 88 лет. Она не только преподносит программирование на Python в увлекательной форме, но и закладывает хорошую базу, которая может быть использована при работе и с другими языками программирования. Подходит для всех, кто хочет научиться программировать, независимо от возраста.

– Бен Омс, инженер-программист в компании Sogeti

Если вы хотите научиться программированию или научить этому ребенка, эта книга – прекрасный выбор.

– Cuberick.com

Очень хорошее введение в программирование для любого, кто хочет развить этот жизненно важный и очень интересный навык.

– Сью Джи, I-Programmer

Уоррен и Картер начинают с самых азов и проходят с детьми или взрослыми весь путь к созданию забавных 2D-игр и симуляторов. Я бы порекомендовал начинающим программистам изучить в первую очередь Python, а эта книга будет прекрасным помощником. Я рекомендую ее своим студентам с момента выхода первого издания.

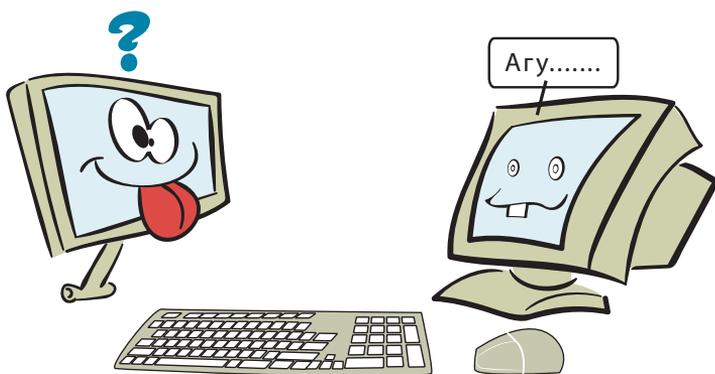
– Дэйв Брикетти,
преподаватель и разработчик программного обеспечения
в компании Dave Briccetti Software LLC

Предисловие

Предисловие – эта та часть в начале книги, которую мы пропускаем, чтобы добраться, наконец, до нужной информации, правильно? Конечно, ты можешь его пропустить, но кто знает, что важного ты не прочитаешь... Оно не очень длинное, поэтому, может быть, все-таки заглянешь?

Что такое программирование?

Если очень кратко, *программирование* означает приказ компьютеру выполнить какое-либо действие. Компьютеры – бестолковые машины. Они не знают, как делать что-либо. Тебе приходится отдавать им команды и учитывать все детали.



Но если ты дашь им верные *инструкции*, они могут сделать много прекрасных и удивительных вещей.

СЛОВАРИК

Инструкция – базовая команда, которую ты даешь компьютеру, обычно для выполнения одного определенного действия.

Компьютерная программа состоит из множества инструкций. Компьютеры выполняют все эти классные вещи сегодня потому, что много умных программистов написали программы, *программное обеспечение*, которое и руководит ими. Программное обеспечение – это программа или набор программ, которые запуска-

ются на твоём или на другом компьютере, к которому ты подключен, например на сервере.

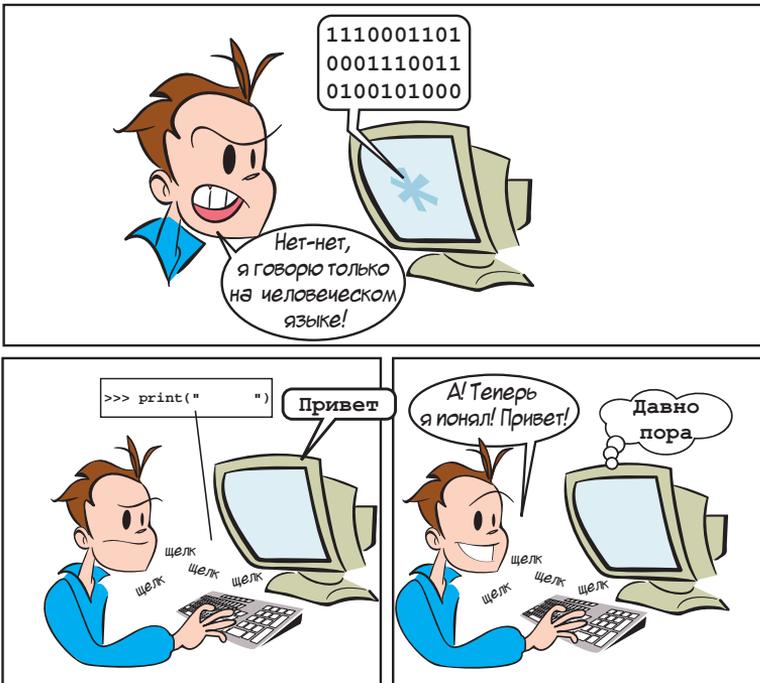
ЧТО ТАМ ПРОИСХОДИТ?



Компьютеры «думают» при помощи многих, многих и МНОГИХ электрических цепей. На самом базовом уровне эти цепи имеют значение ДА или НЕТ. Инженеры и компьютерные специалисты используют значения 0 и 1 вместо ДА и НЕТ. Все эти единицы и нули представляют собой некий код, называемый *двоичным*. Двоичный означает «два состояния». Два состояния – это ДА и НЕТ, или 0 и 1. Запомни: двоичная цифра = бит.

Python – язык для нас и компьютера

Все компьютеры используют двоичный код. Но большинство людей владеет им не очень хорошо. Нам нужен более простой способ донести до компьютера то, что он должен делать. И люди изобрели языки программирования. Компьютерный язык программирования позволяет нам писать команды тем способом, который мы понимаем, а затем переводит их в двоичный код для использования компьютером.



Существует много разнообразных языков программирования. Эта книга научит тебя пользоваться одним из них – Python (читается как «пайтон») – чтобы ты мог заставить компьютер делать то, что тебе нужно.

Чтобы установить подходящую к этой книге версию Python, воспользуйся **специальным инсталлятором Hello World. Мы очень рекомендуем использовать именно его**, чтобы ты установил корректную версию программного обеспечения. Ты найдешь его по ссылке www.dmkpress.com.

Зачем изучать программирование?

Даже если ты не станешь профессиональным программистом (большинство так и не становится), существует несколько причин изучать программирование:

- самая важная: потому что ты этого хочешь! Программирование – очень интересная и полезная вещь как в качестве хобби, так и в виде профессии;
- если ты интересуешься компьютерами и хочешь знать больше о том, как они работают и как заставить их делать то, что тебе надо, то стоит изучать программирование;
- может быть, ты хочешь создать игру или не можешь найти программу, которая делает то, что тебе нужно. Ты сможешь написать ее самостоятельно;
- компьютеры сейчас везде, ты будешь использовать их на работе, в школе и дома. Изучение программирования поможет тебе лучше понимать принципы работы компьютеров.

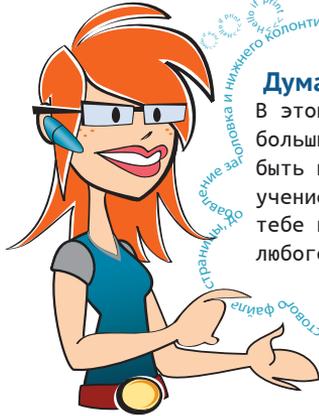
Почему именно Python?

При огромном выборе языков программирования (а их очень много!) почему же мы выбрали именно Python для детской книги? На то есть несколько причин:

- Python прост для изучения. Программы, написанные на нем, легче всего читать, писать и понимать, чем на каком-либо другом языке;
- Python бесплатен. Ты можешь загрузить его – и много-много веселых и полезных программ на нем – бесплатно. Python – программное обеспечение с открытым исходным кодом. То есть любой пользователь может внести свой вклад в его развитие (создать способы, которые позволят выполнять больше задач с его помощью или выполнять те же самые задачи, но проще). Многие программисты внесли свой вклад, и теперь ты можешь загрузить результаты их работы;
- Python – это не игрушка. Хотя он очень хорош для изучения программирования, им пользуются многие профессионалы во всем мире, включая программистов в таких компаниях, как NASA и Google. Поэтому когда ты изучишь Python, тебе не нужно будет переключаться на «настоящий» язык, чтобы писать «настоящие» программы. ты уже будешь уметь это делать;
- Python можно запустить в разных операционных системах. Он доступен для Windows, macOS и Linux. В большинстве случаев программа, написанная на Python, будет работать и в операционной системе Windows, и macOS. Ты можешь использовать примеры из этой книги на любом компьютере, на кото-

ром установлен Python (и помни, что дистрибутив Python ты всегда можешь скачать бесплатно);

- нам нравится Python. Мы получаем удовольствие от изучения и использования этого языка и думаем, что тебе тоже понравится.

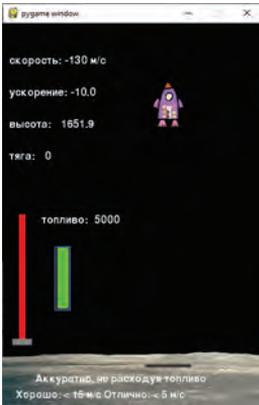


Думай как программист

В этой книге мы изучаем язык Python, но большинство рассматриваемого материала может быть использовано с любым другим языком. Изучение программирования на Python позволит тебе получить отличные навыки для освоения любого другого языка в будущем.

Есть еще кое-что, что мы хотели бы упомянуть...

Один из самых веселых моментов в использовании компьютера (особенно для детей) – игры. Мы научимся делать свои игры, работать с графикой и со звуком. Ниже приведены примеры некоторых программ, которые мы напишем.



Я думаю (по крайней мере, надеюсь), тебе понравится изучать основы языка программирования и писать свои первые программы настолько же сильно, насколько управлять космическими кораблями и лыжниками.

Давай повеселимся!

Благодарности

Благодарности из первого издания

Эта книга никогда не была бы начата и закончена без вдохновения, инициативы и поддержки моей прекрасной жены Патрисии. Когда мы не смогли найти книгу, чтобы удовлетворить интерес нашего сына Картера к программированию, она сказала: «Тебе следует написать ее самому. Это будет отличный проект для вас обоих». Как и в большинстве случаев, она оказалась права. Патрисия умеет раскрывать в людях их лучшие качества. Поэтому мы с Картером начали думать, что будет в этой книге, делать наброски глав и писать примеры программ, искать способы сделать ее забавной и интересной. Раз уж мы взялись за дело, Картер и Патрисия следили за тем, чтобы книга была закончена. Картер отказывался от сказок на ночь, чтобы поработать над ней. А если мы не занимались книгой какое-то время, он напоминал: «Папочка, мы не писали книгу уже несколько дней!» Картер и Патрисия говорили мне, что если ты действительно чего-то хочешь, ты этого добьешься. И все члены нашей семьи, включая нашу дочь Кайру, жертвовали многими часами совместного досуга, пока эта книга не была окончена. Я благодарен им всем за терпение и поддержку, из-за которых эта книга увидела свет.

Написание рукописи – это одно, а передача книги в руки читателя – совсем другое. Эта книга никогда бы не была опубликована без энтузиазма и постоянной поддержки Майкла Стивенса из издательства Manning Publications. С самого начала он «ухватил» идею и согласился в необходимости такого рода издания. Непокколебимая вера Майкла в этот проект и его терпеливое руководство над неопытным писателем были чрезвычайно ценны. Я также хотел бы сказать спасибо всем сотрудникам издательства, которые помогли выпустить эту книгу, а особенно Мэри Пирджис за терпеливую координацию всех производственных моментов.

Эта книга не была бы таковой без веселых и живых иллюстраций Мартина Мертонена. Его работы говорят сами за себя о таланте и художественном вкусе автора. К сожалению, они не показывают, насколько с ним приятно работать. Это было сущее удовольствие.

Однажды я спросил своего друга и коллегу Шона Кавану: «Как бы ты сделал это на языке Perl?» И он ответил: «Никак. Я бы использовал Python». Поэтому я окупился в новый язык программирования. Шон ответил на многие мои вопросы, когда я изучал Python и писал первые программы. Он также собрал инсталлятор с примерами для этой книги. Его помощь неоценима.

Я также хотел бы поблагодарить многих людей, которые просматривали книгу во время написания и помогли работать на рукописью: Вибху Чендрешекар, Пэм Колкхоун, Гордон Колкхоун, д-р Тим Купер, Джош Кроунмейер, Саймон Кроунмейер, Кевин Дрисколл, Джеффри Элкнер, Тэд Феликс, Дэвид Гуджер, Лайза Л. Гудеар, д-р Джон Грейсон, Мишель Хаттон, Хорст Дженс, Энди Джадкис, Кейден Кумар, Энтони Линфант, Шэннон Мэдисон, Кеннет МакДональд, Эван Моррис, профессор Александр Репеннинг, Андре Роберж, Кэрри Дж. Стеллпфлаг, Керби Ернер и Брайан Вейнгаартен.

Также огромное спасибо переводчице книги на русский язык – Александре Калмыковой, и отдельная благодарность редактору русского издания, Михаилу Анатольевичу Райтману.

Конечный результат гораздо лучше благодаря им.

– Уоррен Сэнд

Я хотел бы поблагодарить Мартина Мертонена за его отличные карикатуры на меня, мою маму – за разрешение сесть за компьютер в возрасте двух лет и идею написать книгу, и, самое главное, моего отца за все силы, вложенные вместе со мной в эту книгу, и за то, что он научил меня программированию.

– Картер Сэнд

Благодарности из второго издания

Со вторым изданием книги помогли многие из тех, кто внес свой вклад в первое издание. В дополнение к тем людям, которые были перечислены ранее, мы хотели бы поблагодарить тех, кто помогал со вторым изданием: Бена Умса, Брайана Т. Янга, Коди Роузборо, Дэйва Брикетти, Элизабет Гордон, Айрис Фарэуэй, Мейсона Дженкинса, Рика Гордона, Шона Стебнера и Закари Янга. Спасибо также Игнасио Бельтран-Торресу и Даниэлю Солтису, которые тщательно откорректировали окончательный вариант книги незадолго до ее выпуска.

Мы также хотели бы поблагодарить всех сотрудников издательства Manning, которые помогли сделать это второе издание книги даже лучше, чем первое.

Благодарности из третьего издания

В дополнение к тем людям, которые были перечислены ранее, мы хотели бы поблагодарить тех, кто помогал с третьим изданием: Адайла Ретамала, Бена Макнамара, Бисваната Чоудхури, Бьерна Нейхауса, Боба Даста, Эли Хини, Эвиатара Кафкафи, Джеймса Макгинна, Мэрилин Хурет и Мелиссу Айс.

Мы хотели бы поблагодарить нашего друга Шона Кавана за то, что он в очередной раз создал инсталлятор под Windows для третьего издания.

И еще раз спасибо всем в издательстве Manning, кто помог сделать это третье издание книги даже лучше, чем первые два. Надеюсь, на этот раз нам наконец удалось все сделать правильно!

Об этой книге

Эта книга обучает основам компьютерного программирования. Она предназначена для детей, но ее может использовать любой желающий.

Тебе не нужны никакие начальные знания о программировании для чтения этой книги, но ты должен уметь пользоваться компьютером. Возможно, ты используешь его для чтения почты, поиска во Всемирной паутине, прослушивания музыки, запуска игр или выполнения домашних заданий. Если ты можешь выполнять базовые операции на компьютере, например, запускать программы, открывать и сохранять файлы, у тебя не возникнет проблем с этой книгой.

Что тебе нужно

Эта книга обучает программированию на языке под названием Python. Он бесплатный, его можно скачать из многих источников, в том числе с сайта этой книги. Чтобы научиться программировать с помощью этой книги, тебе потребуется:

- эта книга (конечно же!);
- компьютер под управлением операционной системы Windows, macOS или Linux. Примеры в книге были выполнены на платформе Windows (для получения дополнительных сведений по процессу установки в операционных системах macOS и Linux см. файл *Установка в Linux.pdf* в архиве с примерами для данной книги);
- базовые знания по использованию компьютера (запуск программ, сохранение файлов и т. д.). Если ты испытываешь какие-либо сложности, можно обратиться за помощью к друзьям или родителям;
- разрешение установить Python на компьютер (от родителей, учителя или другого лица, ответственного за него). **Мы настоятельно рекомендуем использовать прилагаемый инсталлятор Hello World**, чтобы установить правильную версию Python, необходимую для этой книги. Ты найдешь его по адресу www.dmkpress.com;
- желание изучать новые вещи, даже если сначала у тебя ничего не получается.

Что тебе не нужно

Чтобы изучать программирование с помощью этой книги, тебе *не* нужно:

- покупать какое-либо программное обеспечение. Все, что тебе понадобится, бесплатно и доступно в архиве с примерами для этой книги по адресу **www.dmkpress.com**;
- знание программирования. Это книга для начинающих.

Изучение книги

Если ты собираешься изучать программирование с помощью этого пособия, ниже представлено несколько подсказок, которые тебе помогут:

- Следи за примерами.
- Набирай код программ.
- Отвечай на проверочные вопросы.
- Не волнуйся, успокойся!

Следи за примерами

Примеры в этой книге выглядят так:

```
if timsAnswer == correctAnswer:
    print("Ты все понял!")
    score = score + 10
```

Старайся следить за примерами и набирать код программ самостоятельно. (Я тебе расскажу подробно, как это делать.) Ты можешь просто сидеть в большом удобном кресле и читать книгу. Возможно, ты даже чему-то научишься. Но ты научишься гораздо большему, если будешь выполнять описываемые действия.

Установка Python

Чтобы получить максимальную пользу от этой книги, тебе нужно установить Python на свой компьютер. **Мы настоятельно рекомендуем использовать инсталлятор Hello World**, чтобы установить правильную версию Python. Установщик Hello World доступен на веб-сайте по адресу **www.dmkpress.com**.

Если ты установишь Python каким-либо другим способом, у тебя может оказаться не та версия Python или необходимые модули. В этом случае ты можешь расстроиться, если твои примеры будут работать не так, как должны.

Набирай код программ

Инсталлятор, который доступен в комплекте с книгой, копирует все примеры программ на твой жесткий диск (если пожелаешь). Он находится на сайте книги: **www.dmkpress.com**. Ты также можешь использовать прилагаемые файлы приме-

ров, но мы рекомендуем набирать их код самостоятельно. Набирая код программ, ты «прочувствуешь» программирование и Python в частности. (И потренируешь свои навыки работы с клавиатурой.)

Отвечай на проверочные вопросы

В конце каждой главы приведены вопросы для закрепления пройденного. Выполняй их все по возможности. Если ты столкнешься с трудностями, попробуй найти кого-то, кто разбирается в программировании, чтобы помочь тебе. Проработайте их вместе, и ты многому научишься. Не подглядывай в ответы, пока не закончишь или пока по-настоящему не зайдешь в тупик. (Да, некоторые ответы есть в конце книги, но, как мы уже сказали, не подглядывай!)

НЕ ВОЛНУЙСЯ, УСПОКОЙСЯ!

Не переживай об ошибках. Более того, совершай их! Мы считаем, что ошибки, их нахождение и исправление – один из лучших способов обучения. В программировании твои ошибки не стоят ничего, кроме времени. Поэтому делай их, учись на них и веселись.



Слово Картеру

Я хотел убедиться, что эта книга подойдет детям, будет веселой и легкой для понимания. К счастью, у меня был помощник. Картер – мой сын, который любит компьютеры и хочет узнать о них больше. Он помогал мне сделать эту книгу правильной. Места, когда он замечал что-то смешное или необычное, мы оформили так:



Отличия третьего издания

Вот что нового в третьем издании по сравнению со вторым:

- в книге теперь используется Python 3 вместо Python 2 (приложение, описывающее различия между Python 2 и Python 3, ты найдешь в конце книги);
- для программирования графического интерфейса в главе 20 мы переключились с PyQt 4 на PyQt 5. PyQt также используется для программы «Висельник» в главе 22 и программы «Тамагочи» в главе 24;
- мы заменили главу 26, в которой рассказывалось о простейшем игровом искусственном интеллекте, новой главой о сетях.

Родителям и учителям

Python – бесплатное программное обеспечение с открытым кодом, нет никакой опасности в установке и использовании его на своем компьютере. Вы можете скачать программное обеспечение – и все необходимое для книги – бесплатно на сайте www.dmkpress.com.

Загруженные файлы легко установить, использовать, они не содержат вирусов и вредоносных программ.

Книги наподобие этой раньше сопровождались дисками с программным обеспечением, но сейчас большинство читателей (и издателей) предпочитают использовать Всемирную паутину.

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте (http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс. Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Приступая к работе

Установка Python

Первое, что необходимо сделать, – установить Python на твой компьютер. **Мы настоятельно рекомендуем использовать инсталлятор Hello World**, чтобы установить правильную версию Python; ты найдешь его на сайте www.dmkpress.com. Выбери версию инсталлятора, которая соответствует операционной системе твоего компьютера.

В СТАРЫЕ ДОБРЫЕ ВРЕМЕНА



На заре эры персональных компьютеров (ПК) все было просто. На многих из них был установлен язык программирования под названием BASIC. Людям ничего не нужно было устанавливать. Требовалось только включить компьютер, и на экране появлялась надпись **READY**¹, после чего можно было начинать набирать программу на языке BASIC. Звучит неплохо, да?

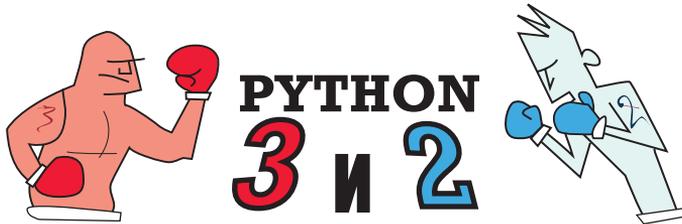
Но эта надпись **READY** была всем, что ты имел в распоряжении. Никаких программ, окон и меню. Если ты хотел заставить компьютер совершить какое-то действие, тебе нужно было писать программу! Не было текстовых редакторов, мультимедийных проигрывателей, браузеров и других вещей, к которым мы привыкли. Не было даже Всемирной паутины. Не было красивой графики и звуков, кроме системного «бип» при совершении ошибки!

¹ Готов (здесь и далее – прим. ред.).

Существуют версии для операционных систем Windows, macOS и Linux. Все примеры в этой книге написаны для операционной системы Windows, но использование Python в macOS или Linux аналогично. Просто следуй инструкциям, чтобы установить правильный дистрибутив для твоей системы.

В этой книге мы используем версию Python 3.7.3. Если ты воспользуешься инсталлятором, содержащимся в архиве с примерами для этой книги, ты тоже получишь эту версию. На момент чтения данной книги могут появиться новые версии Python. Все примеры в книге были проверены в версии 3.7.3. Они должны работать и на более поздних версиях, но мы не можем предвидеть будущее, так что все может случиться.

Если Python уже установлен на твоём компьютере и ты не собираешься пользоваться инсталлятором, тебе нужно убедиться, что некоторые дополнительные компоненты, необходимые для изучения примеров из книги, тоже установлены. См. файл *Установка в Linux.pdf* на сайте www.dmkpress.com, чтобы узнать, как это сделать. Однако повторюсь: чтобы все операции, описанные в книге, получились и у тебя, лучше используй наш инсталлятор, доступный на сайте www.dmkpress.com.



Сравнение Python 3 с Python 2

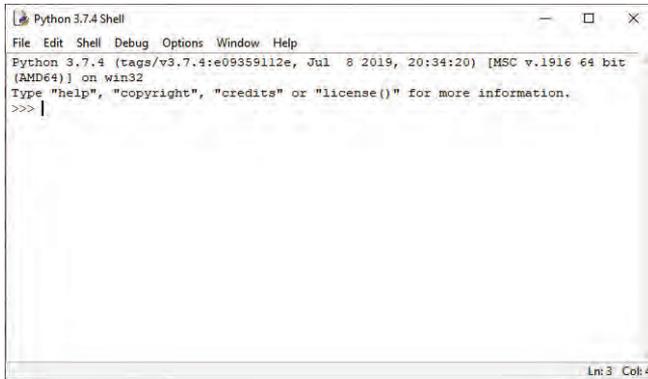
В предыдущих изданиях книги использовался Python 2. С тех пор новая версия программы, Python 3, стала более популярной, поэтому именно ее мы используем в этой книге. Однако оказывается, что Python 3 – на самом деле не столько измененная программа, сколько развилка на дороге. То есть код, написанный на Python 2 (как и пример кода из предыдущих изданий), не всегда будет корректно работать в Python 3, и наоборот.

Для получения более подробной информации о различиях между Python 2 и Python 3 см. приложение Б.

Взаимодействие с Python посредством среды разработки IDLE

Есть несколько способов начать работу с Python. Один из них называется средой разработки IDLE, которой мы сейчас и воспользуемся.

В меню **Пуск** (Start) в папке **Python 3.7** ты увидишь пункт **IDLE (Python 3.7)**. Щелкни мышью по нему – и увидишь открывшееся окно интерпретатора IDLE. Окно будет выглядеть примерно так, как на показано рисунке ниже.



Среда разработки IDLE – это *интерпретатор*¹ Python с графической *оболочкой*. Оболочка – это способ взаимодействия с программой посредством вводимого текста, а эта конкретная оболочка позволяет тебе работать с Python. (Именно поэтому ты видишь слова «Python Shell»² в названии окна.) В среде IDLE есть еще кое-что помимо оболочки, о чем ты узнаешь через минуту.

Символы `>>>` на предыдущем рисунке обозначают *строку приглашения*. Строка приглашения отображается, когда программа ждет от тебя ввода какой-либо команды или инструкции. В данном случае она сообщает о том, что Python готов начать работу.

Введи инструкции, пожалуйста

Давай отправим Python первую инструкцию.

Установи указатель мыши после символов строки приглашения `>>>` и введи код:

```
print(«Привет, мир!»)
```

Нажми клавишу **Enter**. (На некоторых клавиатурах она называется **Return**.) Нажимать клавишу **Enter** необходимо после ввода каждой строки. После ее нажатия ты получишь ответ:

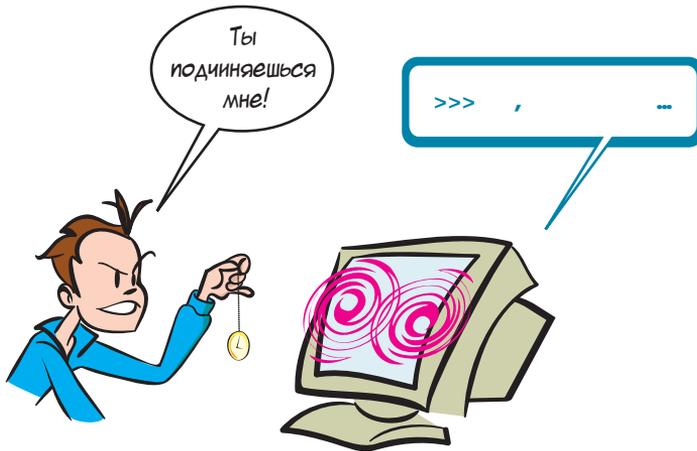
```
Привет, мир!  
>>>
```

¹ Интерпретатор – программа, позволяющая вводить команды на том или ином языке программирования и тут же выполняющая их.

² Оболочка Python (пер. с англ.).

Рисунок ниже показывает, как это выглядит в окне интерпретатора IDLE.

Python сделал именно то, что ты от него потребовал: он написал твоё сообщение. (В программировании команда **print** часто означает отображение текста на экране, а не его печать на бумаге с помощью принтера¹.) Эта строка и есть *инструкция* для Python. Ты уже начал программировать! Компьютер тебе подчиняется!



Кстати, при изучении программирования есть традиция, что первой командой компьютеру должно быть вывести приветствие «Привет, мир!». Ты следуешь традиции. Добро пожаловать в мир программирования!

¹ Print – печать (пер. с англ.).



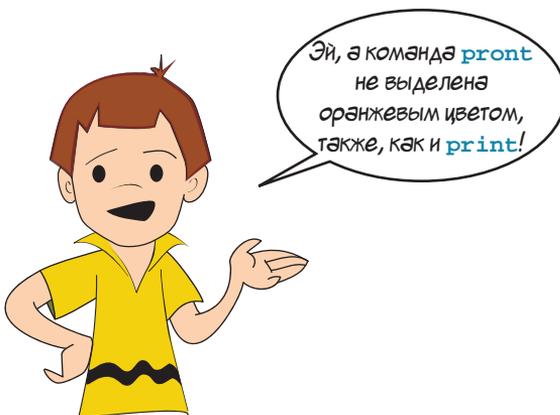
Хороший вопрос! IDLE помогает понять некоторые вещи. Интерпретатор отображает их разным цветом, чтобы ты мог легко различать части кода. (Код – другой термин для инструкций, которые ты даешь компьютеру на языке программирования.) Мы объясним, что же такое эти разные части, далее в этой книге.

Если команда не работает

Если ты сделаешь ошибку, то увидишь сообщение наподобие этого:

```
>>> pront("Привет, мир!")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pront' is not defined
>>>
```

Это сообщение об ошибке означает, что ты ввел команду, которую Python не понимает. В примере выше команда **print** написана неправильно: **pront**, – и Python не знает, что с ней делать. Если случится подобное, попробуй ввести команду снова и убедись, что она набрана правильно.



Все верно. Так произошло потому, что **print** является ключевым словом Python, а **pront** – нет.

СЛОВАРИК

Встроенные функции и ключевые слова – это специальные слова, которые являются частью языка Python. IDLE выделяет их специальным цветом, чтобы ты знал, что они особенные.

Взаимодействие с Python

То, что ты сейчас делал, – это использовал Python в интерактивном режиме. Ты ввел команду (инструкцию), и Python *выполнил* ее немедленно.

СЛОВАРИК

Выполнение команды, инструкции или программы – просто красивое название для ее «запуска» или «включения».

Давай попробуем еще кое-что в интерактивном режиме. Введи следующий код в командной строке:

```
>>> print(5 + 3)
```

Ты должен получить ответ:

```
8
>>>
```

Python может выполнять операции сложения! Это неудивительно, потому что компьютеры отлично знают математику.

Давай попробуем еще:

```
>>> print(5 * 3)
15
>>>
```

В большинстве компьютерных программ и языков символ `*` используется для обозначения умножения. Он называется *звездочкой*.

Если на уроках математики ты привык писать «пятью три» как « 5×3 », тебе нужно привыкать к использованию символа `*` вместо этого. (Символ в верхней части клавиши с цифрой 8 на большинстве клавиатур.)



Хорошо, а как насчет этого:

```
>>> print(2345 * 6789)
15920205
>>>
```



Или насчет этого:

```
>>> print(1234567898765432123456789 * 9876543212345678987654321)
12193263200731596000609652202408166072245112635269
>>>
```



Все верно. С помощью компьютера можно производить действия с очень и очень большими числами. Вот что еще можно сделать:

```
>>> print(«кот» + «пес» )
котпес
>>>
```

Или попробуй вот это:

```
>>> print(«Привет « * 20)
Привет Привет Привет Привет Привет Привет Привет Привет
Привет Привет Привет Привет Привет Привет Привет Привет
```

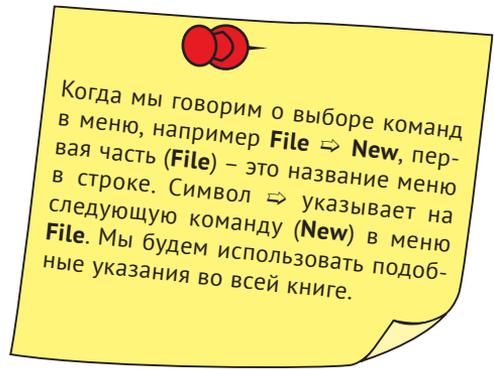
Помимо математики, компьютеры хороши еще и в повторении. Только что мы попросили Python двадцать раз повторить слово «Привет».

Мы еще вернемся к интерактивному режиму позже, а прямо сейчас наступило...

Время программировать

Примеры, которые ты видел до этого, – это простые инструкции (в интерактивном режиме). Они хороши для проверки некоторых возможностей Python, но это еще не программы. Как мы упомянули ранее, программа – это набор инструкций, собранных вместе. Итак, давай напишем свою первую программу на Python.

Для начала нам нужен способ ввода программы. Если ты будешь просто печатать в интерактивном окне, Python ее «не запомнит». Тебе необходимо использовать текстовый редактор (например, «Блокнот» в Windows, TextEdit в macOS или vi в Linux), который позволяет сохранить код программы в файл на жестком диске твоего компьютера. Среда разработки IDLE содержит текстовый редактор, который гораздо удобнее для наших нужд, чем «Блокнот». Чтобы найти его, выбери команду меню **File** ⇒ **New File** в окне интерпретатора IDLE.



Ты увидишь окно, как показано на рисунке ниже. В строке с названием будет указано значение **Untitled**¹, поскольку ты еще не присвоил этому файлу имя.



Теперь введи программу из листинга 1.1 в окно редактора.

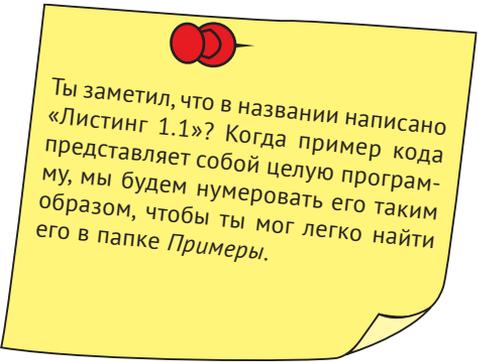
¹ Без названия (пер. с англ.).

Листинг 1.1. Наша первая настоящая программа

```
print («Я люблю пиццу!»)
print ("пицца " * 20)
print ("мямя " * 40)
print ("Я наелся!")
```

Закончив, сохрани программу, выбрав команду меню **File** ⇒ **Save** (Файл ⇒ Сохранить) или **File** ⇒ **Save As** (Файл ⇒ Сохранить как). Присвой файлу имя **pizza.py**. Ты можешь сохранить его в любую папку. Даже создать новую папку для хранения своих программ. Расширение `.py` очень важно, поскольку именно оно сообщает компьютеру, что это программа на языке Python, а не обычный текстовый файл.

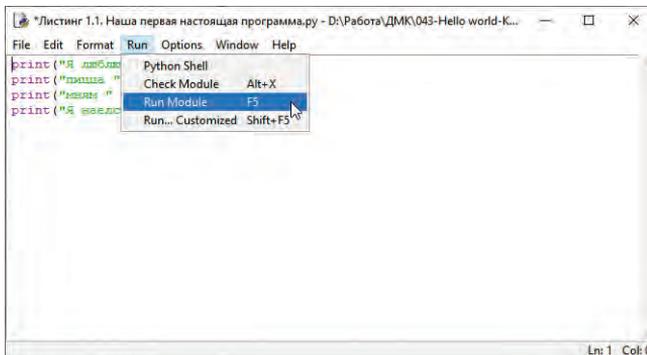
Обрати внимание, что редактор использует разные цвета в программе. Некоторые слова выделены фиолетовым цветом, а некоторые – зеленым. Это происходит потому, что интерпретатор IDLE предположил, что ты будешь набирать программу. Интерпретатор IDLE показывает встроенные функции Python фиолетовым, а значения в кавычках – зеленым цветом. Это упрощает чтение кода программ на языке Python.



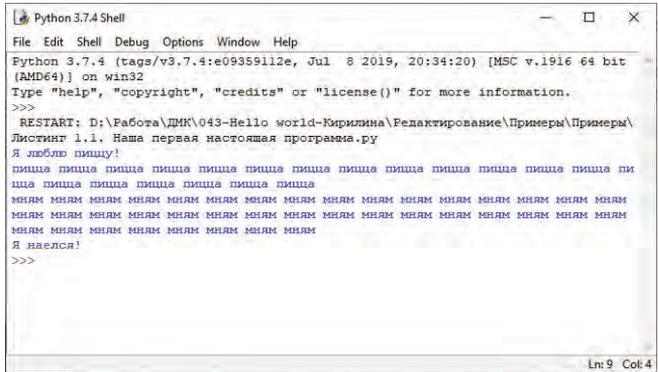
Ты заметил, что в названии написано «Листинг 1.1»? Когда пример кода представляет собой целую программу, мы будем нумеровать его таким образом, чтобы ты мог легко найти его в папке *Примеры*.

Запуск первой программы

Сохранив программу, перейди к меню **Run** (в окне интерпретатора IDLE) и выбери команду **Run Module**, как показано на следующем рисунке. Эта команда запустит твою программу.



Ты увидишь, что окно оболочки Python (первое окно, которое появляется при запуске среды разработки IDLE) снова станет активным, а затем ты увидишь что-то наподобие следующего:



Та часть, где написано слово **RESTART**, говорит о том, что мы запустили программу. (Это помогает ориентироваться, когда ты запускаешь программы одну за другой.)

Потом начинается выполнение программы.

Конечно, она выполнила совсем немного действий. Но ты добился от компьютера выполнения требуемых действий. Наши программы станут более интересными в процессе обучения.

Если что-то пойдет не так

Что случится, если в программе допущена ошибка и она не запустилась? Есть два разных вида ошибок, которые могут случиться. Давай рассмотрим оба вида, чтобы ты знал, что делать в случае каждого из них.

Синтаксические ошибки

Оболочка IDLE выполняет проверку программы перед попыткой запустить ее. Если она находит ошибку, это обычно *синтаксическая ошибка*. Синтаксис – это правописание и грамматические правила языка программирования, то есть синтаксическая ошибка означает, что ты написал нечто неверное в коде. Ниже представлен пример:

```

print («Я люблю пиццу!»)
print («пицца « * 20)
print («мям « * 40)
print(Я наелся!»)
    
```

← Пропущена кавычка

Мы пропустили кавычку между командой **print** и словосочетанием **Я наелся!** Если попробовать запустить эту программу, оболочка отобразит окно с сообщением об ошибке: «SyntaxError» (Синтаксическая ошибка). Тебе потребуется просмотреть свой код и обнаружить ее. Интерпретатор IDLE выделит красным цветом область с ошибкой. Это может быть не совсем точное выделение, но близкое к ошибке.

Ошибки при выполнении

Второй вид ошибок интерпретатор IDLE не может обнаружить до запуска программы. Эти ошибки случаются только при запуске, поэтому их называют *ошибками при выполнении*. Ниже приведен пример подобной ошибки в программе:

```
print(«Я люблю пиццу!»)
print("пицца " * 20)
print("мям " + 40)
print("Я наелся!")
```

Если мы сохраним этот код и попробуем его запустить, программа начнет выполнять инструкции. Первые две строки будут выведены на экран, а затем мы получим сообщение об ошибке:

```
>>>
RESTART: C:/HelloWorld/Примеры/error1.py
Я люблю пиццу!
пицца пицца
пицца пицца пицца пицца пицца пицца пицца пицца пицца пицца
Traceback (most recent call last): ← Начало сообщения об ошибке
  File "C:/HelloWorld/Примеры/error1.py", line 3, in <module> ← Где находится ошибка
    print("мям " + 40) ← Строка с ошибкой в коде
TypeError: must be str, not int ← Возможная ошибка по версии интерпретатора
>>>
```

Строка со словом **Traceback** – начало сообщения. Следующая строка содержит информацию о месте ошибки – имя файла и номер строки. Затем выводится сама строка с ошибочным кодом. Это помогает обнаружить ошибку. В последней части сообщения указано мнение интерпретатора Python о том, что именно было неправильно. Вскоре ты научишься понимать подобные сообщения.



Картер, в Python нельзя складывать разные типы вещей, например текст и число. Поэтому команда `print(«мям» + 40)` спровоцировала ошибку. Похоже на детскую задачу: «Если я возьму 5 яблок и 3 крокодила, сколько получится в итоге?» Ты получишь 8, но 8 чего? Сложение этих вещей вместе не имеет смысла. Но ты можешь умножить почти что угодно на какое-либо число. (Если у тебя есть 2 крокодила, умножив их на 5, ты получишь 10 крокодилов.) Именно поэтому команда `print(«пицца» * 20)` работает.


```

if guess == secret:
    print(«Стой! Ты угадал! Теперь ты знаешь мой секрет!»)
else:
    print(«Больше никаких попыток! Удачи в следующий раз, приятель!»)
    print(«Секретное число», secret)

```

Вывод сообщения в конце игры

Когда ты будешь набирать код, обрати внимание на выделение отступами строк после команды **while** и дополнительное выделение строк после **if** и **elif**. Также не пропусти двоеточия в конце некоторых строк. Если ты правильно используешь двоеточие после нужной строки, редактор сам выделит отступом следующую строку.

Сохранив файл, запусти программу с помощью команды меню **Run** ⇨ **Run Module**, как и в случае с первой программой. Попробуй сыграть в игру. Ниже представлен пример запущенной программы:

```

>>>
RESTART: C:/HelloWorld/Примеры/Листинг_1-2.py
ЯРРР! Я страшный пират Робик, и у меня есть секрет!
Это число от 1 до 100. У тебя есть 6 попыток.
Каков твой ответ? 40
Слишком много, сухопутная крыса!
Каков твой ответ? 20
Слишком много, сухопутная крыса!
Каков твой ответ? 10
Слишком мало, каррамба!
Каков твой ответ? 11
Слишком мало, каррамба!
Каков твой ответ? 12
Стой! Ты угадал! Теперь ты знаешь мой секрет!
>>>

```

Я использовал пять попыток для угадывания числа, которое оказалось 12.

Мы расскажем подробнее об инструкциях **while**, **if**, **else**, **elif** и **input** в следующих главах. Но ты можешь уже сейчас сформировать начальное представление о работе этой программы:

- 1 Секретное число выбирается программой в случайном порядке.
- 2 Пользователь вводит свой вариант числа.
- 3 Программа сравнивает догадку с секретным числом: больше или меньше?
- 4 Пользователь продолжает угадывать, пока у него не закончатся попытки.
- 5 Когда догадка совпадет с секретным числом, игрок выигрывает.



Что ты узнал?

Ух! Ты узнал много нового. В этой главе ты:

- установил Python;
- научился запускать среду разработки IDLE;
- узнал об интерактивном режиме;
- предоставил интерпретатору Python некоторые инструкции и выполнил их;
- увидел, как Python выполняет арифметические действия (включая работу с очень большими числами!);
- запустил текстовый редактор IDLE, чтобы набрать код своей первой программы;
- запустил свою первую программу на Python!
- узнал о сообщениях об ошибке;
- запустил свою вторую программу на Python: игру на угадывание чисел.

Проверь свои знания

- Как запустить интерпретатор IDLE?
- Какое действие выполняет команда `print`?
- Какой символ служит для представления операции умножения на языке Python?
- Что отображает интерпретатор IDLE при запуске программы?
- Каким другим словом можно назвать запуск программы?

Попробуй самостоятельно

- 1 В интерактивном режиме интерпретатора Python подсчитай количество минут в неделе.
- 2 Напиши короткую программу для вывода трех строк: твое имя, дата твоего рождения и твой любимый цвет. Результат должен выглядеть примерно так:

```
Меня зовут Петя Иванов.  
Я родился 1 марта 1998 года.  
Мой любимый цвет — синий.
```

Сохрани программу и запусти ее. Если программа не выполняет ожидаемые действия или ты получаешь сообщение об ошибке, попробуй исправить ошибки в коде и повторить попытку.