



Майк МакГрат




- ЦВЕТНОЕ ИЗДАНИЕ
- ЧЕТКИЕ ПОШАГОВЫЕ ИНСТРУКЦИИ
- СОЗДАЙТЕ ПРОГРАММУ С НУЛЯ
- ОПЫТ В РАЗРАБОТКЕ НЕ НУЖЕН

# ПРОГРАММИРОВАНИЕ ДЛЯ НАЧИНАЮЩИХ



ОСНОВНЫЕ ПРИНЦИПЫ

издание **2**



**БОМБОРА**  
ИЗДАТЕЛЬСТВО

Москва

УДК 004.42  
ББК 32.973.2-018  
М15

CODING FOR BEGINNERS:  
MODERN PROGRAMMING FOR ALL AGES

2nd Edition  
by Mike McGrath

Copyright © 2022

Translated and reprinted under a licence agreement from the Publisher: In Easy Steps, 16 Hamilton Terrace,  
Holly Walk, Leamington Spa, Warwickshire, U.K. CV32 4LY



**МакГрат, Майк.**

М15 Программирование для начинающих : основные принципы / Майк МакГрат ; [перевод с английского В. В. Краснянской]. — 2-е издание. — Москва : Эксмо, 2025. — 192 с. — (Программирование для начинающих).

ISBN 978-5-04-213835-5

Открывать новое бывает непросто, но с «Программированием для начинающих» вы легко погрузитесь в мир ИТ даже при отсутствии опыта. Следуйте понятным инструкциям, выполняйте интересные практические задания. Внутри есть вся необходимая информация, рассмотрены ключевые концепции разработки и подобраны наглядные цветные примеры. Вы научитесь: писать программный код на Python, хранить информацию в структурах данных и создавать алгоритмы в рамках объектно-ориентированного программирования. Смело воплощайте свои идеи в код!

УДК 004.42  
ББК 32.973.2-018

ISBN 978-5-04-213835-5

© Краснянская В. В., перевод на русский язык, 2025  
© Оформление. ООО «Издательство «Эксмо», 2025

## 1

### Начнем!

9

Программный код	10
Установка и настройка Setting Up	12
Знакомство с IDLE	14
Справочная служба	16
Сохранение программ	18
Хранение переменных	20
Добавление комментариев	22
Правила именования	23
Краткие итоги	24

## 2

### Сохранение данных

25

Хранение данных ввода	26
Управление данными на выходе	27
Распознавание типов данных	28
Преобразование типов данных	30
Игра-угадайка	32
Исправление ошибок	34
Краткие итоги	36

## 3

### Выполнение операций

37

Выполнение арифметических действий	38
Присваивание значений	40
Сравнение значений	42
В поисках истинности	44
Проверка условий	46
Порядок выполнения действий	48
Краткие итоги	50

## 4

### Работа со списками

51

Создание списков	52
Изменение списков	54
Неизменяемые списки	56
Список с уникальными значениями	58
Именованные элементы списка	60
Краткие итоги	62

## 5

## Блоки управления

63

Ветвление и выборы	64
Циклы со счетчиком	66
Циклы с условием	68
Пропуск итерации	70
Ловим ошибки	72
Краткие итоги	74

## 6

## Создание функций

75

Объявление функций	76
Добавление параметров	78
Возврат результатов	80
Сохранение функций	82
Импорт функций	84
Краткие итоги	86

## 7

## Алгоритмы сортировки

87

Сортировка с копированием	88
Сортировка выбором	90
Сортировка вставками	92
Сортировка пузырьком	94
Сортировка слиянием	96
Сортировка с разделением	98
Краткие итоги	100

## 8

## Импорт библиотек

101

Python в подробностях	102
Математические расчеты	104
Вычисления с десятичными дробями	106
Который час?	108
Запуск таймеров	110
Краткие итоги	112

## 9

## Управление текстом

113

Обработка строк	114
Форматирование строк	116
Изменение строк	118
Доступ к файлам	120
Обработка содержимого файлов	122
Обновление содержимого файлов	124
Краткие итоги	126

## 10

### Программные объекты

127

Объявление классов	128
Копирование экземпляров классов	130
Обращение к свойствам классов	132
Наследование классов	134
Переопределение методов	136
Действовать схожим образом	138
Краткие итоги	140

## 11

### Создание интерфейсов

141

Запуск интерфейсов	142
Реагирующие кнопки	144
Вывод сообщений	146
Сбор входных данных	148
Список опций	150
Радиокнопки	152
Флажки или чекбоксы	154
Добавление изображений	156
Краткие итоги	158

## 12

### Разработка приложений

159

Генерация случайных чисел	160
Планирование потребностей	162
Разработка компоновки виджетов	164
Работа со статикой	166
Загрузка динамических данных	167
Добавление функционала	168
Тестирование программ	170
Внедрение приложений	172
Краткие итоги	174

## 13

### Расширяя навыки

175

Что такое компиляторы	176
Компиляция кода	178
Код на C	180
Код на C++	182
Код на C#	184
Код на Java	186
Краткие итоги	188

### Предметный указатель

189

# Как пользоваться этой книгой

Эта книга дала мне, Майку МакГрату, уникальную возможность — познакомить с разработкой программ читателей, у которых нет никакого опыта программирования. Это издание предназначено для начинающих, но выходит за рамки минимального набора понятий, поэтому, возможно, некоторые темы легче понять после того, как у читателя появится некоторый опыт разработки простых программ, перечисленных в книге. В примерах иллюстрируются характерные особенности работы с кодом на примере популярного языка программирования Python.

## Условные обозначения, принятые в книге

Чтобы приведенный в каждом примере код выглядел понятным на каждом шаге, я применил систему цветowych условных обозначений, которая по умолчанию используется в редакторе кода Python. Ключевые слова языка программирования выделены оранжевым цветом, названия стандартных функций — фиолетовым, функции, названия которым дал разработчик, — синим, текстовые строки — зеленым, комментарии — красным, а весь остальной код — черным:

```
# Функция для вывода на экран приветствия
def greet( reader ) :
    print( 'Welcome to Coding for Beginners in easy steps' , reader )
```

Дополнительно для того, чтобы обозначить каждый описанный в пошаговых инструкциях файл с исходным кодом, на полях рядом с ним указаны значки и имена файлов (см. илл. справа).



program.py

## Скачивание исходного кода

Для удобства я поместил в один zip-архив все файлы с исходным кодом из примеров, приведенных в этой книге. Вы можете скачать весь архив, следуя этим простым шагам:

- 1 Пройдите по адресу [www.ineasysteps.com](http://www.ineasysteps.com), затем перейдите на [Free Resources](#) (Бесплатные ресурсы) и выберите раздел [Downloads](#) (Загрузки).
- 2 Найдите в списке название книги [Coding for Beginners in easy steps, 2nd edition](#), перейдите по ссылке и, чтобы загрузить архив, нажмите на ссылку [All code examples](#) (Все примеры кода).
- 3 Затем распакуйте содержимое архива в любом удобном месте на вашем компьютере.

На приведенных в книге скриншотах показаны реальные результаты выполнения приведенного в шаге кода. Если у вас не получился результат, описанный в примере, просто сравните ваш код с образцом из файла, который вы загрузили, и найдите, где ошиблись.

# 1 Начнем!

<i>Добро пожаловать в удивительный мир программирования!</i>	<b>10 Программный код</b>
<i>В этой главе рассказывается о том, как создать свою собственную среду разработки и написать первую программу.</i>	<b>12 Установка и настройка</b>
	<b>14 Знакомство с IDLE</b>
	<b>16 Справочная служба</b>
	<b>18 Сохранение программ</b>
	<b>20 Хранение переменных</b>
	<b>22 Добавление комментариев</b>
	<b>23 Правила именования</b>
	<b>24 Краткие итоги</b>



## Программный код

Компьютер — это всего лишь машина, которая может очень быстро обрабатывать простой набор команд. Такой набор инструкций и называется программой, а сами инструкции — кодом.

Людей, которые пишут компьютерные программы, называют программистами или разработчиками. Именно их работа делает компьютеры полезными практически в каждой сфере современной жизни.

- **В руках** — компьютеры находятся в мобильных устройствах, чтобы мы могли общаться с помощью голосовой связи, текстовых сообщений и социальных сетей.
- **В доме** — компьютеры находятся в таких бытовых приборах, как телевизоры, игровые приставки и стиральные машины.
- **В офисе** — компьютеры находятся в настольных устройствах, служащих для обработки текстов, составления зарплатных ведомостей и графического дизайна.
- **В магазине** — компьютеры находятся в устройствах для розничной торговли, таких как автоматические кассовые аппараты и сканеры штрихкодов.
- **В автомобиле** — компьютеры находятся в устройствах управления, отвечающих за работу двигателя, антиблокировочную систему и безопасность.
- **В небе** — компьютеры находятся в самолетах, осуществляя пилотирование, а также в диспетчерских центрах, управляющих воздушным движением.

Это еще далеко не все примеры того, как компьютеры влияют на современную жизнь. Но на самом деле компьютеры очень глупы! Они совершенно не могут думать сами.

Компьютер — это всего лишь набор электронных компонентов, которые обычно называют «аппаратным обеспечением» или «железом». Чтобы компьютер что-то делал, ему нужно выдать программные инструкции, то есть «программное обеспечение».

Важно, чтобы у каждой машины были четкие пошаговые инструкции, которые компьютер может выполнить без ошибок. Таким образом, разработчик должен разбить задачу на простые недвусмысленные шаги. Например, программа движения мобильного робота из помещения на улицу должна включать в себя инструкции, позволяющие роботу обнаружить дверной проем и обойти любые препятствия. Поэтому разработчик должен всегда принимать во внимание все возможные неожиданные трудности, с которыми может столкнуться программа.

Программные инструкции следует предоставлять компьютеру на языке, который он может понять. На самом простом уровне компьютер способен

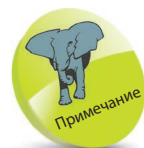
воспринимать «машинный код», перемещающий элементы памяти так, чтобы машина могла выполнить поставленные задачи. С этим неудобочитаемым низкоуровневым кодом очень утомительно работать, поскольку для того, чтобы компьютер справился с самой простой работой, требуется множество строк инструкций.

К счастью, с течением времени появилось большое количество высокоуровневых языков программирования, позволяющих разработчику составлять инструкции в более понятной для человека форме. Компилятор или интерпретатор автоматически переводит эти современные высокоуровневые программы в машинный код, понятный для компьютера. Чтобы стать разработчиком, вам нужно изучить по крайней мере один из перечисленных ниже высокоуровневых языков программирования:

- **C** — мощный компилируемый язык программирования, точно отражающий машинный код и использующийся для разработки операционных систем.
- **C++** — усовершенствованный компилируемый язык программирования, разработанный на основе C с поддержкой классов для объектно-ориентированного программирования (ООП).
- **C#** — современный компилируемый язык программирования, разработанный компанией Microsoft для программной платформы .NET Framework и спецификации общезыковой инфраструктуры (Common Language Infrastructure, CLI).
- **Java** — переносимый компилируемый язык, разработанный для того, чтобы работать на любой платформе независимо от архитектуры аппаратного обеспечения.
- **Python** — динамически интерпретируемый язык программирования, допускающий как функциональное, так и объектно-ориентированное программирование.

У всех естественных языков, на которых люди говорят между собой, можно найти определенное сходство — например, в них есть глаголы и существительные. Так и в языках программирования имеются схожие особенности: в каждом из них имеются «структуры данных» для хранения информации и «структуры управления», определяющие то, как программа выполняется.

В этой книге я привожу примеры, показывающие, как писать компьютерные программы, используя язык программирования Python, поскольку он имеет простой синтаксис, не требует компиляции, включает мощную библиотеку стандартных функций и может использоваться как для консольных программ, так и для многооконных приложений с графическим пользовательским интерфейсом.



Программы, написанные на интерпретируемом языке, можно запустить сразу, но программы на компилируемых языках вначале должны быть скомпилированы.



В целом Python — это продукт, поставляемый в полном комплекте, по принципу «батарейки прилагаются».



Инструменты для установки Python на macOS и других платформах также бесплатно доступны на [python.org/downloads](https://python.org/downloads).



После добавления Python в системную переменную Path язык программирования становится доступным из любого каталога. Завершив установку, вы можете ввести в командной строке команду `python -V` и увидеть ответ интерпретатора и его версию.

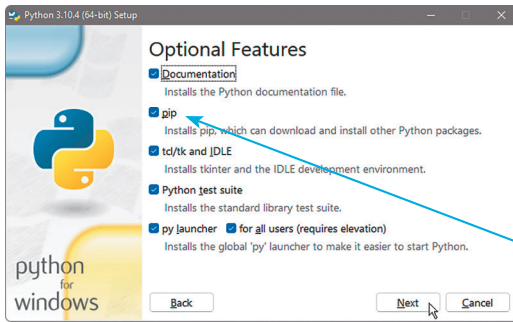
# Установка и настройка

Прежде чем вы сможете писать программы на Python, необходимо установить на компьютер среду разработки, в том числе интерпретатор Python и стандартную библиотеку модулей кода, поставляемую вместе с интерпретатором. Все это можно загрузить бесплатно на странице Python Software Foundation.

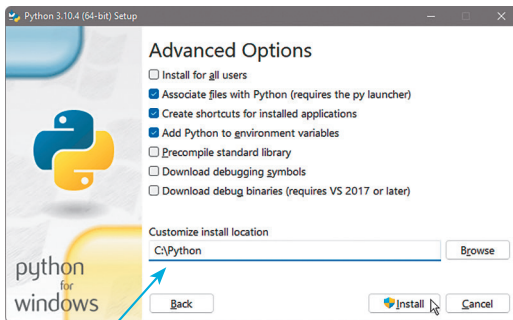
- 1 Запустите браузер и перейдите на страницу [python.org/downloads](https://python.org/downloads), затем нажмите кнопку **Download** (Загрузить), чтобы загрузить последнюю версию для вашей операционной системы. В данном примере мы используем инструмент установки для Windows из файла под названием `python-3.10.4-amd64.exe`.
- 2 Когда загрузка закончится, найдите в папке Downloads (Загрузки) исполняемый файл с расширением `.exe` и запустите его, в результате чего появится диалоговое окно установки.



- 3 Далее не забудьте отметить опцию **Add Python 3.10 to PATH** (Добавить Python 3.10 в системную переменную PATH).
- 4 Теперь нажмите **Customize installation** (Настроить параметры установки), чтобы открыть диалоговое окно **Optional Features** (Дополнительные функции) и выбрать необходимые параметры.
- 5 Нажмите **Next** (Далее), чтобы открыть диалог дополнительных параметров **Advanced Options**, и выберите там **Associate files** (Ассоциированные файлы), **Create shortcuts** (Создавать ярлыки) и **Add Python to environment variables** (Добавить Python к переменным среды).



Убедитесь, что вы установили менеджер управления пакетами **pip**. Далее в книге мы будем им пользоваться, чтобы установить пакет **PyInstaller**, с помощью которого вы сможете создавать распространяемые приложения.

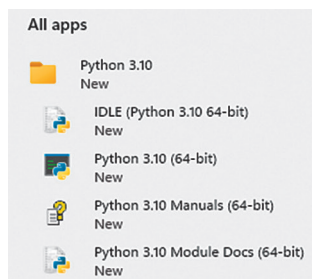


**6** Выберите место установки в каталог **C:\Python**, затем нажмите кнопку **Install** (Установить).

**7** Если вам предложат, выберите опцию **Disable path length limit** (Отключить ограничение длины пути), чтобы избежать каких-либо проблем с путями.

**8** Чтобы завершить установку Python, нажмите кнопку **Close** (Закреть).

После завершения установки в меню Start/All Apps (Пуск/Все приложения) добавится группа Python. Самое важное приложение в этой группе — **IDLE** (Python Integrated DeveLopment Environment), запускающее интегрированную среду разработки Python.



Вы часто будете пользоваться IDLE, так что нажмите на ярлык правой кнопкой мыши и выберите «Дополнительно → Закрепить на панели задач», чтобы программу было проще найти на рабочем столе Windows.

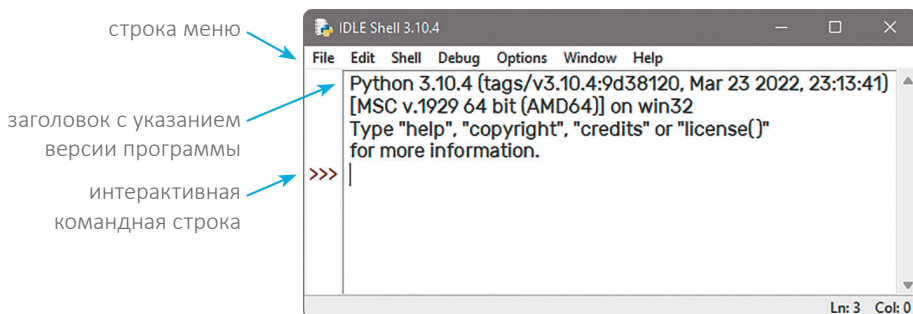


## Знакомство с IDLE

В установленный пакет программного обеспечения Python входит среда разработки Integrated DeveLopment Environment (IDLE), где вы с легкостью можете писать код и запускать программы или фрагменты кода, написанные на Python. IDLE фактически состоит из двух частей, предназначенных для программной разработки:

- Интерактивная оболочка Shell.
- Редактор кода.

После запуска IDLE открывается новое окно, где вы можете видеть строку меню, заголовок, где указана версия ПО, и строку `>>>`. Это интерактивная оболочка Shell, где можно взаимодействовать непосредственно с интерпретатором Python, вводя команды в строку.



Если интерпретатор поймет строку, которую вы ввели, он даст соответствующий ответ, в ином случае — сообщит об ошибке.

Интерпретатор может вывести строку текста, если использовать в командной строке функцию Python с именем **print()**. Текст вводится внутри скобок со знаками кавычек с обеих сторон.

Также интерпретатор может вывести результаты простых арифметических действий, если ввести в командную строку правильное выражение.

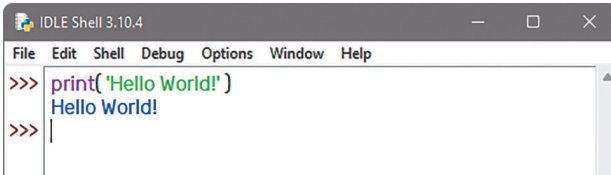
Если же выражение будет неправильным, например таким, где число пытаются поделить на 0, интерпретатор выдаст ошибку, где будет подробно описана суть возникшей проблемы.



В большинстве языков программирования текстовые строки требуется заключать в кавычки, чтобы отделить от программного кода. Обычно Python-разработчики используют одинарные кавычки.

1 Откройте интерактивную командную строку и точно введите в нее следующее выражение:  
`print( 'Hello World!' )`

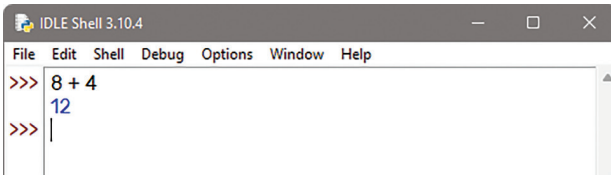
2 Далее нажмите на клавишу **ввода**, чтобы увидеть ответ интерпретатора.



```
File Edit Shell Debug Options Window Help
>>> print( 'Hello World!' )
Hello World!
>>> |
```

3 Теперь введите в командную строку выражение суммы:  
`8 + 4`

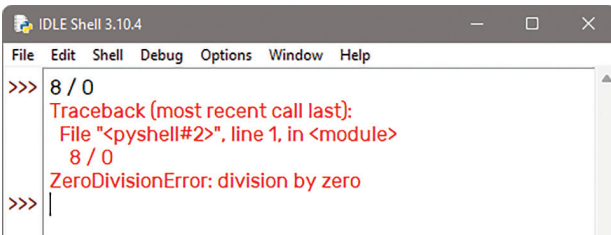
4 Нажмите **ввод**, и вы увидите, что интерпретатор выведет результат.



```
File Edit Shell Debug Options Window Help
>>> 8 + 4
12
>>> |
```

5 Введите в командную строку неверное выражение:  
`8 / 0`

6 Нажмите **ввод**, и интерпретатор выведет сообщение об ошибке.



```
File Edit Shell Debug Options Window Help
>>> 8 / 0
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    8 / 0
ZeroDivisionError: division by zero
>>> |
```



Пробелы в выражениях игнорируются. С таким же успехом вы можете ввести выражение `8+4` без пробелов.



Интерактивная оболочка Shell чаще всего используется для проверки фрагментов кода.



## Справочная служба

В интерактивной оболочке Shell имеется прекрасная справочная служба, где Python-разработчик может найти информацию по любой теме, связанной с программированием на Python. Вызвать эту справочную службу можно, написав в командной строке `>>> help()`. Появится приветственное сообщение, и строка будет начинаться с `help>`, указывая, что программа теперь находится в справочном режиме.

- 1 Откройте интерактивную оболочку Shell и точно введите в командную строку следующее сообщение:  
`help()`
- 2 Далее нажмите клавишу **ввода**, чтобы перейти в справочный режим.



Приветственное сообщение справочной службы также содержит полезные советы, но сейчас мы не будем на них останавливаться для краткости.

```

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
>>> help()
welcome to python 3.10's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the internet at https://docs.python.org/3.10/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics". Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

Ln: 48 Col: 0
  
```

- 3 Теперь введите в командную строку название следующей темы:  
`keywords`
- 4 Нажмите **ввод**, и вы получите список всех ключевых слов языка программирования Python.

```

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
help> keywords
Here is a list of the Python keywords. Enter any keyword to get more help.

False          class           from            or
None           continue       global          pass
True           def            if              raise
and            del            import          return
as             elif           in              try
assert         else           is              while
async          except         lambda          with
await          break          finally         nonlocal       yield
break          for            not

help> |
Ln: 54 Col: 6
  
```



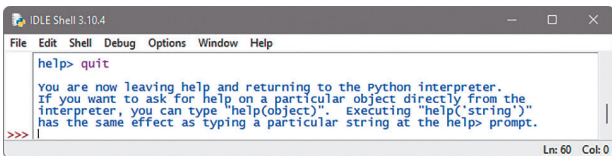
Ключевые слова — это словарь языка программирования. Заметим, что в Python регистр имеет значение при вводе ключевых слов. Все они пишутся со строчной буквы, за исключением `False`, `None` и `True`.

5

Теперь введите в командную строку справочной службы слово **quit**.

6

Нажмите клавишу **ввода**, чтобы выйти из справочной службы, и вернитесь в командную строку интерактивной оболочки Shell.



```

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
help> quit
You are now leaving help and returning to the python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)". Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>>
Ln: 60 Col: 0

```

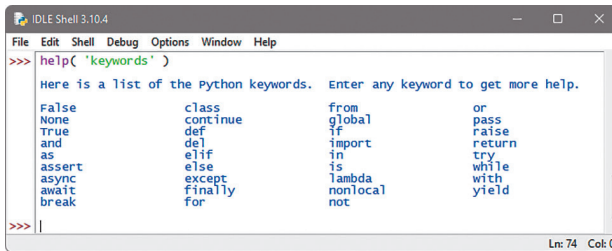
Если у вас возник вопрос по какой-то конкретной теме, вы можете просто ввести ее название без кавычек внутри скобок в выражении **help()** в командной строке.

7

Точно введите это выражение в командную строку:  
**help( 'keywords' )**

8

Нажмите клавишу **ввода**, и вы получите список всех ключевых слов языка программирования Python. При этом вы останетесь в командной строке интерактивной оболочки Shell.



```

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
>>> help( 'keywords' )
Here is a list of the Python keywords. Enter any keyword to get more help.
False          class          from           or
None           continue     global         pass
True           def          if            raise
and            del         import        return
as            elif        in           try
assert        else       is           while
async        except    lambda       with
await        finally  nonlocal    yield
break        for      not
>>>
Ln: 74 Col: 0

```



После инструкции quit круглые скобки не нужны, поскольку это команда справочной службы, а не выражение языка программирования Python.



Ключевые слова имеют особое значение в языке программирования. Их нельзя использовать для названия объектов в коде.