

Что такое функция

Функция — это блок кода, который можно многократно вызывать на выполнение. Она является фундаментальной частью любого языка программирования.

Функция позволяет разделять программу на самостоятельные, но связанные части. Программисты используют функции, чтобы сделать программу модульной и избежать повторения кода.

Функция может использоваться для обработки данных, она получает на вход значения, обрабатывает его и возвращает результат в программу. Также она может не возвращать значение, а выводить его на экран или записывать в файл.

Программист может написать собственную функцию или использовать готовые решения языка, если они есть, конечно. Например, лучше самому не написать функцию для определения максимального числа, а воспользоваться стандартной `max()`.

Объявление

Объявляя функцию, нужно следовать определенным правилам:

- ✓ Объявление происходит с помощью ключевого слова `def`, за ним идёт имя функции и круглые скобки `()`.
- ✓ Аргументы, передаваемые в функцию, должны находиться в круглых скобках. Там же можно определить их значения по умолчанию, указав их после знака равно.
- ✓ Тело функции начинается после знака двоеточия. Важно не забыть об отступах.
- ✓ Чтобы выйти из функции в Python, используют оператор `return [значение]`. Если оператор опущен, будет возвращено значение `None`.

Синтаксис объявления

Параметры (аргументы) нужно передавать в том порядке, в котором они определены в круглых скобках.

```
def Имя(аргументы):  
    "Документация"  
    Тело (инструкции)  
    return [значение]
```

Ключевое слово `def` в начале функции сообщает интерпретатору о том, что следующий за ним код — есть её определение. Всё вместе — это объявление функции.

```
# объявим функцию my_function()
def my_function():
    # тело функции
```

Аргументы часто путают с параметрами:

- **Параметр** — это переменная, которой будет присваиваться входящее в функцию значение.
- **Аргумент** — само это значение, которое передается в функцию при её вызове.



Параметры и аргументы функции

```
# a, b - параметры функции
def test(a, b):
    # do something

# 120, 404 - аргументы
test(120, 404)
```

Ключевая особенность функций — возможность возвращать значение

Для этого используется слово `return`. Предположим, вы часто умножаете числа. Вы не осведомлены заранее, целые они или вещественные, но хотите, чтобы результат был целым всегда. Решим задачу с помощью функции:

```
# она будет принимать два множителя, а возвращать их округленное
# до целого числа произведение
def int_multiple(a, b):
    product = a * b
    # возвращаем значение
    return int(product)

print(int_multiple(341, 2.7))
```

Достаточно часто возникают вопросы: "почему моя функция ничего не возвращает?!", "почему из функции возвращается None?", "не могу понять откуда появляется None...".

Для начала необходимо понимать и помнить, что любая функция в Python всегда что-то возвращает и если не используется оператор `return` для возврата значения (такие случаи бывают, но об этом позднее), то функция возвращает объект `None`. В случае если `return` используется, но после него ничего не указывается явно, то по умолчанию считается, что там стоит объект `None`.

```
# Не используем оператор return, поэтому результат не возвращается.
def func(x):
    x * x

print(func(10))
#=> None

# Используем оператор return, но не задаем явное значение/используем некорректно
def func(x):
    x * x
    return

print(func(10))
#=> None

# Корректно возвращаем результат с использованием оператора return
def func(x):
    return x * x

print(func(10))
#=> 100
```

<https://pythonchik.ru/osnovy/funkcii-v-python>