

Разработка программного обеспечения на языке Python

[Обзорная панель](#) ▶ [Мои курсы](#) ▶ [Разработка ПО на языке Python](#) ▶ [Программирование на языке Python](#) ▶

[Лекция 4. Процедурное программирование. Понятие функции](#)

Лекция 4. Процедурное программирование. Понятие функции

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

Функции. Часть 1



ФУНКЦИИ

ПРИМЕР

необязательные параметры

00:00 / 03:19



Функции представляют блок кода, который выполняет определенные операции и который можно повторно использовать, например, как это было с функцией `print()`, когда мы хотели вывести значения на консоль.

Помимо функции `print` Python имеет множество других встроенных функций, а также при необходимости мы можем определить свои функции.

Синтаксис определения функции:

```
def имя_функции ([параметры]):  
    инструкции
```

Чтобы определить свою функцию необходимо использовать выражение **def**, после мы указываем название функции, в скобках параметры функции, которые могут и отсутствовать. А на новой строке, с обязательным отступом от начала строки мы пишем инструкции. Например, напомним функцию, которая будет складывать 2 числа и выводить результат на консоль.

```
def Summa (x,y):  
    z = x + y  
    print(z)
```

Но если запустить программу, то на консоли мы ничего не увидим, потому что функции необходимо вызвать. Для вызова функции указывается имя функции, после которого в скобках идет передача значений для всех ее параметров.

```
Summa(2,3)
```

```
C:\Users\Анастасия\PycharmProjects'
```

```
5
```

Обратите внимание, что функция сначала определяется, а потом вызывается.

Некоторые параметры функции мы можем сделать необязательными, указав для них значения по умолчанию при определении функции. Например, в этом примере к числу, которое будет передаваться в качестве параметра функции, всегда будет прибавляться единица.

```
def Summa(x, y = 1):
    z = x + y
    print(z)
```

```
Summa(2)
```

```
C:\Users\Анастасия\PycharmProjects'
```

```
3
```

Если функция имеет несколько параметров, то необязательные параметры должны идти после обязательных.

В примерах выше при вызове функции значения передаются параметрами функции по позиции. Но также можно передавать значения параметрам по имени. Для этого при вызове функции указывается имя параметра и ему присваивается значение:

```
def Summa(x, y):
    z = x + y
    print(z)
```

```
Summa(y = 3, x = 2)
```

```
C:\Users\Анастасия\PycharmProjects
```

```
5
```

С помощью символа звездочки можно определить параметр, через который можно передавать неопределенное количество значений. Это может быть полезно, когда мы хотим, чтобы функция получала несколько значений, но мы точно не знаем, сколько именно. Например, пусть наша функция суммы в качестве параметров будет принимать неопределенное количество значений.

Как вы можете увидеть, теперь мы можем вызывать функцию Summa с различным количеством параметров.

```
def Summa(*numbers):
    z = 0
    for n in numbers:
        z += n
    print(z)
```

```
Summa(3,2)
```

```
Summa(1,4,8)
```

```
Summa(3,7,6,33,2)
```

```
C:\Users\Анастасия\PycharmProjects
```

```
5
```

```
13
```

```
51
```

В нашем примере функция ничего не возвращает, она просто выводила на консоль результат.

Но функция может и возвращать результат своих операций. Для этого в функции используется оператор **return**, после которого указывается возвращаемое значение. Изменим знакомую уже нам функцию `Summa`, добавив в нее оператор **return**.

```
def Summa(*numbers):  
    z = 0  
    for n in numbers:  
        z += n  
    return z  
  
print(Summa(3,2))  
C:\Users\Анастасия\PycharmProjects  
5
```

Теперь функция возвращает значение, которое мы потом уже выводим на консоль.

Оператор **return** не только возвращает значение, но и производит выход из функции. Поэтому он должен определяться после остальных инструкций.

Функции. Часть 2

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◀ [Задание 6. Загранпаспорт](#)

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

[Задание 7. Среднеарифметическое](#) ▶

© 2010-2023 Центр обучающих систем
Сибирского федерального университета, sfu-kras.ru

Разработано на платформе moodle
Beta-version (3.9.1.5.w3)

[Политика конфиденциальности](#)

[Соглашение о Персональных данных](#)

[Политика допустимого использования](#)

Контакты +7(391) 206-27-05
info-ms@sfu-kras.ru

[Скачать мобильное приложение](#)

[Инструкции по работе в системе](#)