



GeekBrains

# Основы Python



**GeekBrains**

Урок 3

# Функции

# На этом уроке

1. Именные функции.
2. Оператор return.
3. Аргументы функций.
4. Анонимные функции.
5. Ещё раз о встроенных функциях.
6. Функция range для многократно выполняемых действий.
7. Области видимости переменных в функциях.
8. Документирование кода функций.
9. Алгоритм создания функции.

# Именованные функции

Функция определяется с помощью инструкции **def**,  
после которой следует имя функции.

```
def my_sum(arg_1, arg_2):  
    res = arg_1 + arg_2
```


# Оператор return

Определяет выход из функции и передачу  
результата в точку вызова функции

```
def my_sum(arg_1, arg_2):  
    return arg_1 + arg_2
```


# Аргументы функций

## Позиционные




```
def first_func(var_1, var_2, var_3):  
    print(first_func(10, 20, 30))
```

## Именованные




```
def second_func(var_2, var_1, var_3):  
    second_func(var_1=10, var_2=20, var_3=30)
```

## Обязательные



```
def first_func(var_1, var_2, var_3):  
    print(first_func(10, 20, 30))
```

## Необязательные



```
def second_func(var_1, var_2=20, var_3=30):  
    print(second_func(10))
```

# Анонимные функции

Содержат только одно выражение.

выполняются быстрее именных

```
my_func = lambda p_1, p_2: p_1 + p_2  
print(my_func(2, 5))
```



7

# Еще раз о встроенных функциях

## Операции с символами

Функция	Описание
<b>ord()</b>	Принимает Unicode-символ и возвращает соответствующий код (целое число)
<b>chr()</b>	Принимает целое число и возвращает Unicode-символ, соответствующий переданному числу (коду)
<b>len()</b>	Принимает любой объект-последовательность (строка, набор байтов, список, кортеж) или объект-коллекцию (словарь, множество) и возвращает число элементов последовательности



# Ещё раз о встроенных функциях

## Математические функции

Функция	Описание
<b>abs()</b>	Принимает целое число или число с плавающей точкой. Возвращает абсолютное значение числа (по модулю)
<b>round()</b>	Принимает число с плавающей точкой. Округляет число до ближайшего целого числа. Может принимать число знаков после запятой, до которых необходимо выполнить округление
<b>divmod()</b>	Принимает два числа, возвращает также два числа (частное и остаток от деления чисел)
<b>pow()</b>	Принимает два числа. Позволяет возвести первое число в указанную степень
<b>max()</b>	Принимает итерируемый объект и возвращает самый большой элемент
<b>min()</b>	Принимает итерируемый объект и возвращает наименьший элемент
<b>sum()</b>	Суммирует элементы последовательности

# Функция range()

Отвечает за генерацию набора чисел в пределах  
указанного диапазона.

```
print(list(range(7))) # целые числа в диапазоне [0, 7)
print(list(range(2, 8))) # целые числа в диапазоне [2, 8)
print(list(range(1, 9, 2))) # целые числа в диапазоне [1, 9) с шагом 2
print(list(range(1, -7, -2))) # целые числа в диапазоне [1, -7) с шагом -2
print(list(range(0))) # целые числа в диапазоне (0, 0)
print(list(range(1, 0))) # целые числа в диапазоне (1, 0)
```

# Зоны видимости переменных

Локальная



Глобальная

Нелокальная

# Документирование кода функций

Однострочное



```
"""Возвращает путь до директории."""
```

Многострочное



```
"""Возвращает частное от деления.  
  
Именованные параметры:  
p_1 -- делимое (по умолчанию 0.0)  
p_2 -- делитель (по умолчанию 0.0)  
  
"""
```

# Алгоритм создания функции

1. Придумать информативное имя функции.
2. Подготовить строки документации: назначение функции, типы данных параметров, тип данных результата.
3. Придумать информативные имена параметров функции.
4. Написать тело функции.



# ИТОГИ

1. Научились представлять логику программ в виде блоков-инструкций (функций).
2. Научились передавать в функцию параметры, выполнять её вызов и возвращать результат.
3. Познакомились с анонимными функциями и зоной видимости.
4. Узнали о полезных встроенных функциях и об алгоритме создания собственных функций.