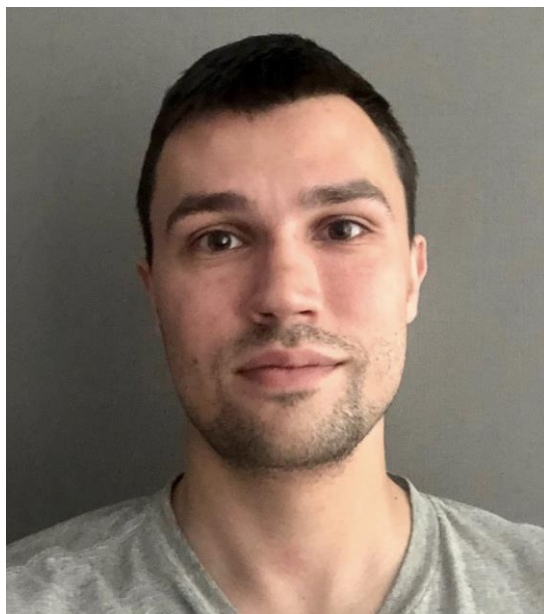


# Python Starter

Функции часть 2

# Python Starter

## Introduction



Бондаренко Кирилл

Data scientist & Python developer



<https://www.linkedin.com/in/kirill-bond/>



<https://medium.com/@bond.kirill.alexandrovich>



# Python Starter

Тема урока

Функции часть 2

# Python Starter

## План урока

1. Область видимости функции
2. Вложенные функции
3. Стандартные функции в python
4. Анонимные функции
5. Рекурсия
6. Примеры работы с функциями

# Python Starter

## Область видимости функции

**Область видимости** – это область, в которой интерпретатору доступны ваши переменные, функции и прочее. Область видимости бывает локальная и глобальная.

Переменные, которые объявлены вне функции, называются **глобальными переменными** в Python. Эти переменные имеют большую область видимости и доступны всем функциям, которые определены после объявления глобальной переменной.

Когда переменная объявляется внутри функции, то это локальная переменная. Область действия локальной переменной ограничена функцией, в которой она создается. Это означает, что она доступна в функции, где она объявлена, но не вне этой функции.

# Python Starter

## Область видимости функции

```
>>> # global
... x = 1
>>> def print_x():
...     print(x)
...
>>> print_x()
1
>>>
```

Глобальная область видимости

```
>>> # local
... def print_local_y():
...     y = 1
...     print(y)
...
>>> print_local_y()
1
>>> y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'y' is not defined
>>>
```

Локальная область видимости

# Python Starter

## Вложенные функции

Функции можно объявлять и вызывать внутри друг друга.

Это может быть поле, если вы хотите чтобы функция была доступна только в локальной области видимости другой функции.

```
>>> def external():  
...     def internal():  
...         return 1  
...     print(internal())  
...  
>>> external()  
1  
>>> internal()  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'internal' is not defined  
>>> 
```

# Python Starter

## Стандартные функции в python

В Python существует множество встроенных функций, которые можно и нужно использовать для упрощения вашего кода.

**print** – функция для вывода сообщения в консоль.

**len** – функция, которая возвращает длину объекта.

```
>>> print('hello')
hello
>>>
>>> len([1, 2, 3])
3
>>> len('abc')
3
>>> 
```



# Python Starter

## Стандартные функции в python

**Функции преобразования типов** – нужны для того, чтобы можно было создавать новые объекты определенного типа.

**str** – преобразование в строку

**int** - преобразование в целое число

**float** – преобразование в не целое число

**list** - преобразование в список

**dict** – преобразование в словарь

**set** – преобразование в множество

**tuple** – преобразование в кортеж

**bool** – преобразование в True/False

```
>>> str(1)
'1'
>>> int(1)
1
>>> float(12)
12.0
>>> list('123')
['1', '2', '3']
>>> dict([
... ('key', 'value')
... ])
{'key': 'value'}
>>> set([1,2,3,3,3])
{1, 2, 3}
>>> tuple('123')
('1', '2', '3')
>>> bool(1)
True
>>>
```

# Python Starter

## Стандартные функции в python

**sum** – возвращает сумму набора чисел.

```
>>> sum([1,2,3])  
6
```

**min** и **max** – возвращает минимум и максимум из набора чисел соответственно.

```
>>> min([1, 2, 3])  
1  
>>> max([1, 2, 3])  
3  
>>>
```

# Python Starter

## Анонимные функций

**Анонимные функции** – используются в случае, если нам их нужно куда-то передать и больше мы не будем их использовать повторно. Такая запись будет компактнее. У такой функции не будет имени.

Синтаксис:

**lambda** x: x

Где первый x – это параметр, который передается в функцию, а второй – то, что возвращается.

Эквивалент обычной функции:

```
def some_name(x):  
    return x
```

```
>>> my_list = [1, 2, 5, 3, 4]  
>>> my_list.sort(key=lambda x: -x)  
>>> my_list  
[5, 4, 3, 2, 1]  
>>>
```

# Python Starter

## Рекурсия

**Рекурсия** – это когда функция вызывает сама себя. В рекурсии обязательно должна быть конечная глубина для того, чтобы не было закливания.

```
>>> def factorial(n):  
...     if n == 0:  
...         return 1  
...     else:  
...         return n * factorial(n - 1)  
...  
>>> factorial(5)  
120  
>>> 
```

# Python Starter

## Примеры работы с функциями

1. Напишите функцию, где дано натуральное число  $n$ . Выведите все числа от 1 до  $n$ .
2. Напишите функцию для просчета площади прямоугольника, в параметрах передаются высота и длинна.

# Проверка знаний

TestProvider.com



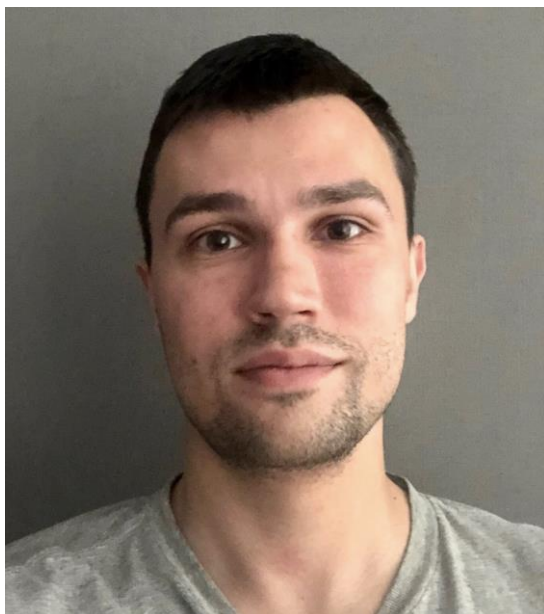
Проверьте как Вы усвоили данный материал на [TestProvider.com](http://TestProvider.com)

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

# Python Starter

Спасибо за внимание! До новых встреч!



**Бондаренко Кирилл**

Data scientist & Python developer



# Информационный видеосервис для разработчиков программного обеспечения

