

ДЕТСКАЯ ОНЛАЙН-ШКОЛА ПРОГРАММИРОВАНИЯ

# HELLO WORLD



#ШПАРГАЛОЧКИ



# ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

## Начальный уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити в Telegram: [https://t.me/hw\\_school](https://t.me/hw_school)



# Функции



**Функция** — это твоя собственная подпрограмма, выполняющая определенный набор команд. Чтобы создать функцию, нужно написать ключевое слово **def** (define - определить) и имя функции после него. Имя функции должно соответствовать тем же правилам, что и имя переменной.

После имени функции обязательно нужно поставить круглые скобки, а в них перечислить **параметры**, которые должна **принимать функция**. Если функция ничего не принимает - скобки будут пустыми.

После скобок пишем двоеточие (:) - прямо как после условий и циклов. А на следующей строке, с отступом в **4 пробела** можно писать код, который функция должна выполнять.



В функцию можно не только **передать** значение, но и вернуть из нее какие-то данные в место **вызова**. Для этого используется команда **return**. Важно - **return** заканчивает работу функции, поэтому код, написанный после, работать не будет.

Чтобы функция заработала, ее нужно **вызвать**. Для этого достаточно написать имя функции в нужном месте программы и поставить круглые скобки после него. Важно - если при создании функции было указано, что она принимает какие-то **параметры**, то при вызове ей обязательно нужно передать соответствующие этим параметрам **значения**.



**Лямбда-функция** - это специальная функция без имени, которую можно записать в переменную. Чтобы создать такую функцию, нужно написать ключевое слово **lambda**, перечислить через запятую параметры, которые она принимает (тут скобки не нужны) и поставить двоеточие (:). После двоеточия мы пишем какое-то действие (лямбда-функция выполняет только одно действие) и его результат будет возвращен в программу. Команда **return** тут не нужна.

Пример создания и вызова лямбда-функции, записанной в переменную **mul**, которая **принимает** числа **a** и **b** и **возвращает** их произведение:

```
mul = lambda a, b: a * b
```

```
mul(5, 7) # 35
```



`def имя_функции ():`

Создание функции

`return`

Команда, возвращающая значения из функции, также прекращает её работу

`def mult (a, b):`

Создание функции mult, которая принимает два параметра

`mult = lambda a, b: a * b`

Создание лямбда функции и запись её в переменную mult