

Содержание

Функция	2
Задания по функциям	5
Аргументы функции	6
Задания по аргументам	7
Команда return	8
Задания по return	9



Функция

Функция – это такой кусок кода, который можно вызывать по имени

чтобы не писать одно и то же много раз!

```
def ИМЯ_ФУНКЦИИ() :  
    КОД_ФУНКЦИИ
```

Например:

```
def palundra() :  
    print("Палундра!")  
    print("AAAAAAAAAAAAAAAAAA")
```

Функции сами по себе ничего не делают и не выполняют

Для этого функцию необходимо вызвать по имени

```
def palundra() :  
    print("Палундра!")  
    print("AAAAAAAAAAAAAAAAAA")  
  
palundra()
```

Не забыть про скобки!

Выводит следующее

```
C:\Users\dima2\AppData\Local\Programs  
Палундра!  
AAAAAAAAAAAAAAAAAA  
  
Process finished with exit code 0
```

Теперь вызовем несколько раз

```
def palundra():  
    print("Палундра!")  
    print("AAAAAAAAAAAAAAAA")  
  
palundra()  
palundra()  
palundra()
```

Теперь выводит так

```
C:\Users\dima2\AppData\Local\Programs  
Палундра!  
AAAAAAAAAAAAAAAA  
Палундра!  
AAAAAAAAAAAAAAAA  
Палундра!  
AAAAAAAAAAAAAAAA  
  
Process finished with exit code 0
```

Без функции нам бы пришлось самим все это одно и то же прописывать!

Можно использовать переменные!

```
def plus():  
    print("x сейчас равен", x)  
  
x = 1  
plus()  
x = 2  
plus()  
x = 3  
plus()
```

Вывод

```
C:\Users\dima2\AppData\Local\Programs\  
x сейчас равен 1  
x сейчас равен 2  
x сейчас равен 3  
  
Process finished with exit code 0
```

Чтобы **менять переменные, объявленные вне функций**, используем слово **global**

```
global ИМЯ_ПЕРЕМЕННОЙ
```

Пример:

```
def plus():  
    global x  
    x = x + 1  
    print("Прибавил единицу: x =", x)  
  
x = 1  
plus()  
plus()  
plus()
```

```
C:\Users\dima2\AppData\Local\Programs\  
Прибавил единицу: x = 2  
Прибавил единицу: x = 3  
Прибавил единицу: x = 4  
  
Process finished with exit code 0
```



Задания по функциям

1. Создай функцию `hi()`, которая выводит на экран твое имя и **вызови ее 5 раз**

Пример: функция `hello()`, которая выводит приветствие, с вызовом 3 раза

```
def hello():  
    print("Ну превед..")  
  
hello()  
hello()  
hello()
```

```
C:\Users\dima2\AppData\Local\Program  
Ну превед..  
Ну превед..  
Ну превед..  
  
Process finished with exit code 0
```

2. Создай переменную `Y = 2` и функцию `Umnoz()`, которая умножает `Y` на 2 и выводит ее

Пример: функция `Deli()`, которая делит `X` на 2 и выводит ее

```
def Deli():  
    global X  
    X = X / 2  
    print("Поделил X на 2 = ", X)  
  
X = 400  
Deli()  
Deli()  
Deli()
```

```
C:\Users\dima2\AppData\Local\Program  
Поделил X на 2 = 200.0  
Поделил X на 2 = 100.0  
Поделил X на 2 = 50.0  
  
Process finished with exit code 0
```

Аргументы функции

Функция может принимать **данные на вход**

Эти данные называются **аргументами** или **параметрами функции**

Аргументы перечисляются в скобках через запятую

```
def kozel(name):  
    print(name, " козел!")  
  
kozel("Физрук")  
kozel("Гриша")
```

```
C:\Users\dima2\AppData\Local\Programs\Python\Python39-32\python.exe  
Физрук козел!  
Гриша козел!  
  
Process finished with exit code 0
```

Функция с двумя аргументами a и b

```
def plus(a, b):  
    print("Считаю...")  
    print("a + b =", a + b)  
    print("a - b =", a - b)  
    print()  
  
plus(5, 2)  
plus(-7, 9)
```



```
C:\Users\dima2\AppData\Local\Programs\Python\Python38-32\python.exe
Считаю...
a + b = 7
a - b = 3

Считаю...
a + b = 2
a - b = -16

Process finished with exit code 0
```

Задания по аргументам

1. Создай функцию, которая пишет гадость (или что-то еще) про того, чье имя будет ее аргументом

Пример есть на странице 6

2. Создай функцию, которая выводит результат умножения и деления двух аргументов

Пример тоже есть на странице 6

Команда return

В функциях можно не только писать гадости и считать всякую шляпу, но и **возвращать результат** командой **return**

```
return РЕЗУЛЬТАТ
```

Пример:

```
def plus(x):  
    return x + 1  
  
print(plus(1))
```

В функцию передали 1, а вернула она 2

```
C:\Users\dima2\AppData\Local\Progra  
2  
  
Process finished with exit code 0  
|
```

Функция проверки пароля:

```
def check(password):  
    if (password == "1234"):  
        return True  
    else:  
        return False  
  
print(check("555"))  
print(check("1234"))
```

```
C:\Users\dima2\AppData\Local\Progra  
False  
True  
  
Process finished with exit code 0
```

Задания по return

1. Сделай функцию `square(x)`, которая возвращает квадрат числа. Выведи результат работы функции от аргументов 1, 3, 5

Пример: функция `kub(a)`, которая возвращает куб аргумента `a`, и выводит результаты от аргументов 3, 2 и 1

```
3 usages
1 def kub(a):
2     return a * a * a
3
4 print(kub(3))
5 print(kub(2))
6 print(kub(1))
```

Run циклы x

C:\Users\dima2\AppData\Local\Programs

27
8
1

Process finished with exit code 0

2. Сделай функцию `max_number(a, b)`, которая возвращает большее число

Пример: функция `min_number(x, y)`, которая возвращает меньшее число из двух аргументов `x` и `y`

```
1 def min_number(x, y):
2     if (x > y):
3         return x
4     else:
5         return y
6
7 print(min_number(x: 3, y: 6))
8 print(min_number(x: 1, y: 1))
9 print(min_number(x: 999, y: 2))
```

Run циклы x

C:\Users\dima2\AppData\Local\Programs

6
1
999

Process finished with exit code 0

3. Сделай функцию `proverka(password)`, которая проверяет равен ли аргумент `password` твоему паролю (пароль придумай сам!) и возвращает `True`, если пароль верный, иначе – возвращает `False`

Пример: функция `proverka(password)` проверяет равно ли введенное значение с клавиатуры паролю “димасик”

```
1  def proverka(password):
2      if password == "димасик":
3          return True
4      else:
5          return False
6
7  пароль = input("Введи пароль:")
8  print(proverka(пароль))
```

C:\Users\dima2\AppData\Local\Program
Введи пароль:qqqqqq
False
Process finished with exit code 0

