УЧЕБНЫЙ ЦЕНТР №1

МОДУЛЬ 1
ТЕМА 3

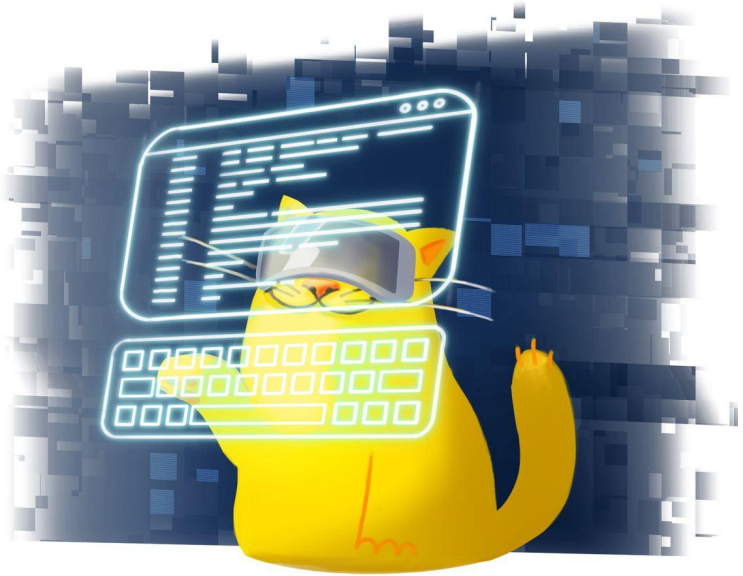
Занятие №1

«Функции. Передача параметров, возвращение результата»

О чем будем говорить сегодня

Тема: функции, передача параметров, возвращение результата

Цель занятия: изучить команды по созданию функций и передаче параметров



Результаты работы

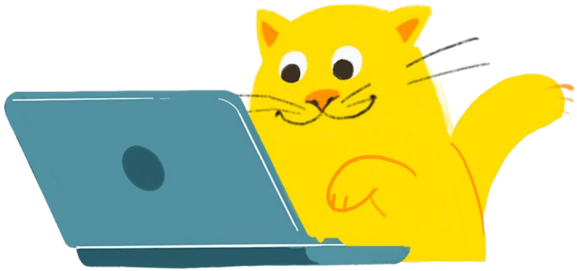
- Научимся создавать свои функции
- Научимся определять аргументы функций
- Научимся разделять глобальные и локальные переменные

Это нужно знать

Функция – часть программного кода, к которой можно обратиться из другого места программы.
Аргументы функции – компоненты функции.

Самые применяемые функции:

- встроенные
- пользовательские (свои)



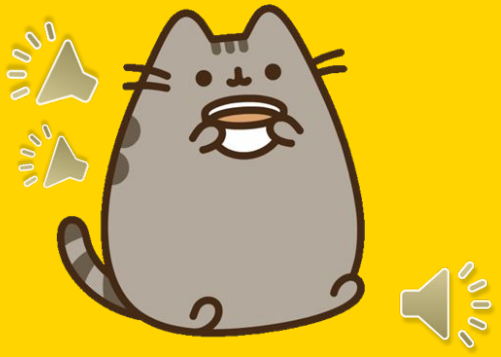
Что помните с прошлого раза?

1. Какую роль выполняет цикл **while** в языке *Python*?
2. Что такое **оператор прерывания цикла**?
3. Как можно передавать аргументы циклу?
4. Какую роль выполняет ключевое слово **def** в языке *Python*?
5. Как можно использовать оператор **and** в языке *Python*?

1С
УЧЕБНЫЙ ЦЕНТР №1

Функции

Учимся писать свои функции



Немного теории: функции

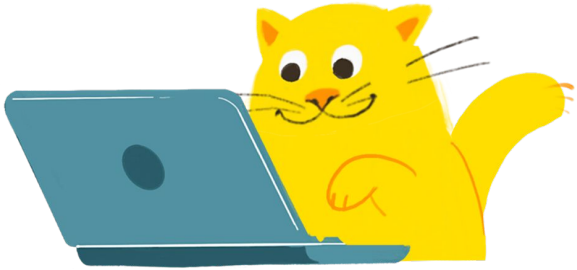
Функции способны работать с аргументами и возвращать результат. Объявить **функцию** можно при помощи слова **def**.

Пример использования: функции

Функция вычисляет сумму двух чисел

```
1.py  third.py  07.08.py
1   # объявление функции
    1 usage
2   def sum():
3       # инструкции
4       x = 10
5       y = 15
6       # вывод результата
7       print('Сумма:', x+y)
8   # вызов функции
9   sum()

07.08
  .exe C:\Users\user\PycharmProjects
  \pythonProject3\07.08.py
Сумма: 25
```



Начинаем практику: функции

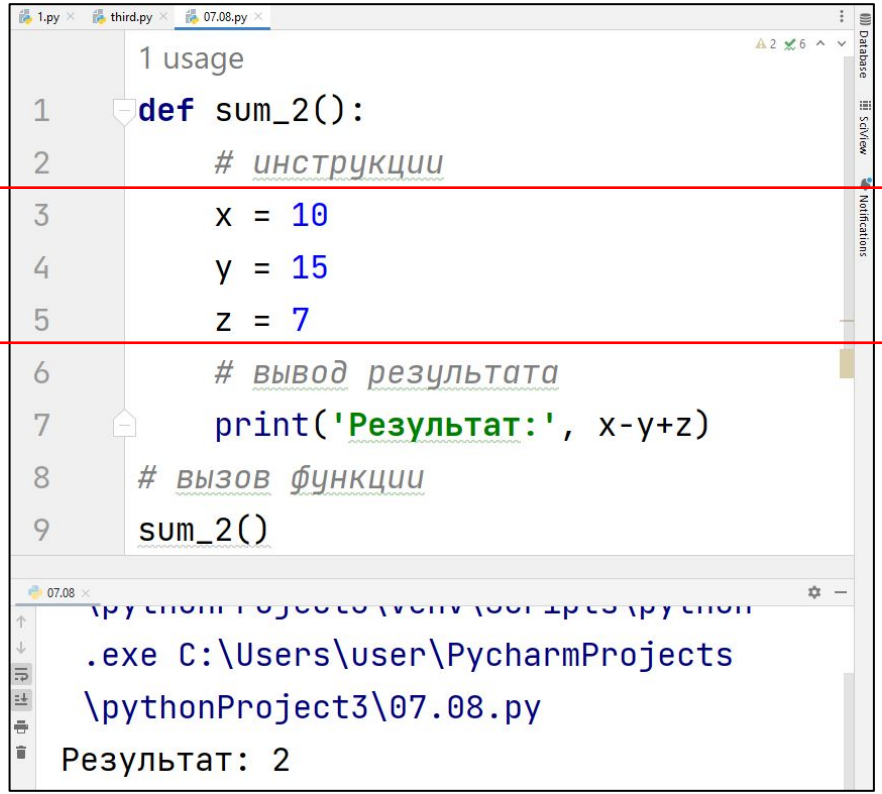
Задание.

Посмотрите на код функции и предположите - что она выведет?

```
1 usage
def sum_2():
    # инструкции
    x = 10
    y = 15
    z = 7
    # вывод результата
    print('Результат:', x-y+z)
# вызов функции
sum_2()
```

Время на выполнение - 5 минут

Посмотрим на код

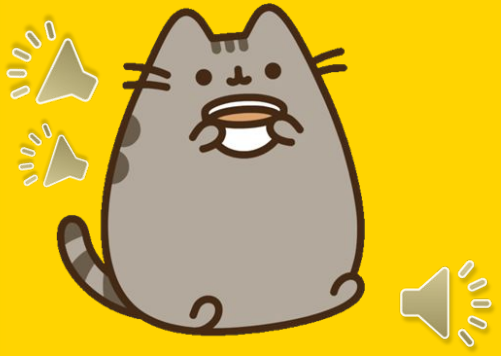


Здесь функция работает с переменными **‘x’**, **‘y’** и **‘z’**

1С
УЧЕБНЫЙ ЦЕНТР №1

Аргументы функции

Знакомимся с аргументами



Немного теории:
аргументы функции

Функция может принимать произвольное количество аргументов или не принимать их совсем:

```
1 usage
def func_3(x, s):
    # вывод результата
    print('Результат:', x+s)
# вызов функции
func_3( x: 10,  s: 15)
```

```
C:\Users\user\PycharmProjects
 \pythonProject3\venv\Scripts\python
 .exe C:\Users\user\PycharmProjects
 \pythonProject3\07.08.py
Результат: 25
```

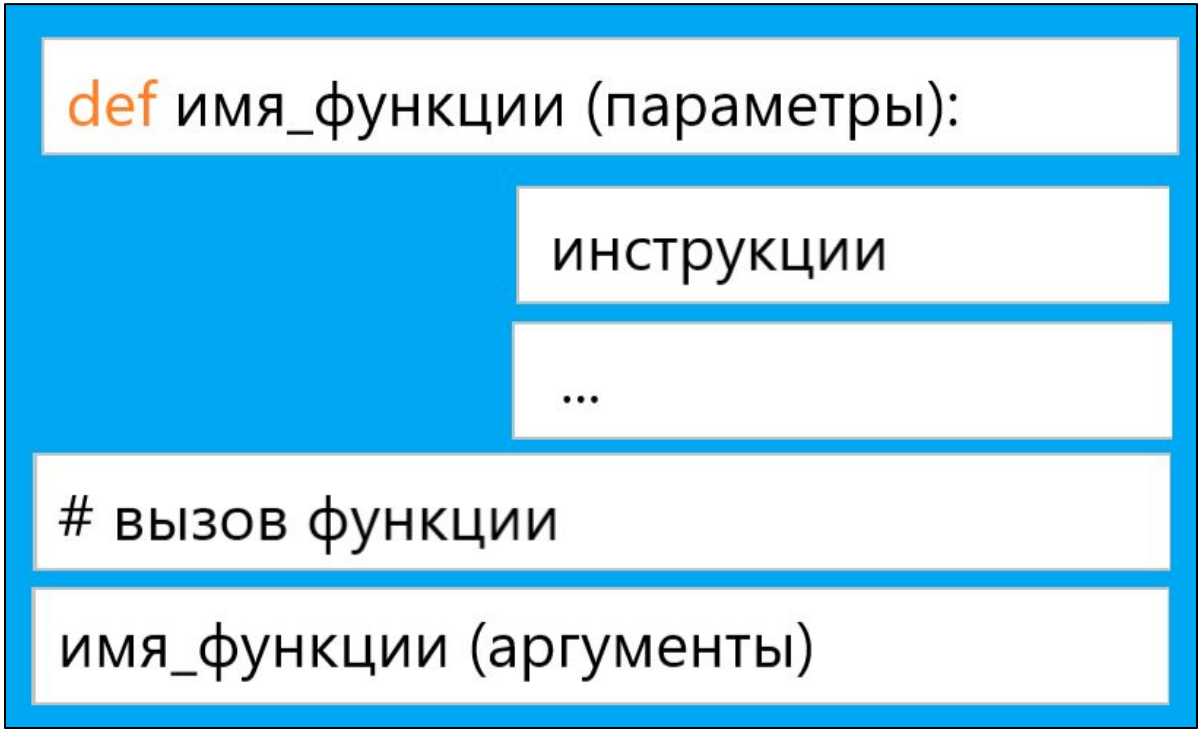
```
1 usage
def func_3():
    # инструкции
    x = 10
    s = 15
    # вывод результата
    print('Результат:', x+s)
# вызов функции
func_3()
```

```
.exe C:\Users\user\PycharmProjects
\pythonProject3\07.08.py
Результат: 25
```

Немного теории:

аргументы функции

Схема работы функции:



Начинаем практику:
аргументы функции

Задание.

Создайте функцию, в которой возвращается произведение переменных **‘q’, ‘w’** и **‘e’** с типом данных ‘**int**’.
Значения переменных указываются при вызове функции.

Время на выполнение - 5 минут

Решение задания

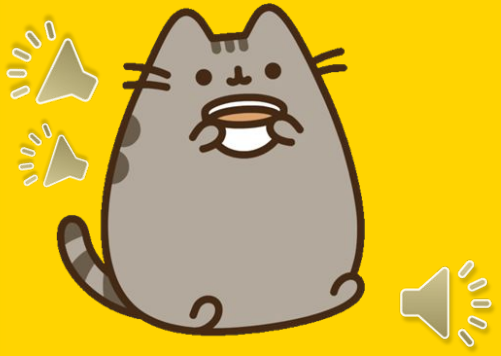
```
1 usage
def func_5(q, w, e):
    # вывод результата
    print('Результат:', q*w*e)
# вызов функции
func_5( q: 5,  w: 6,  e: 7)
```

```
C:\Users\user\PycharmProjects
 \pythonProject3\venv\Scripts\python
 .exe C:\Users\user\PycharmProjects
 \pythonProject3\07.08.py
Результат: 210
```

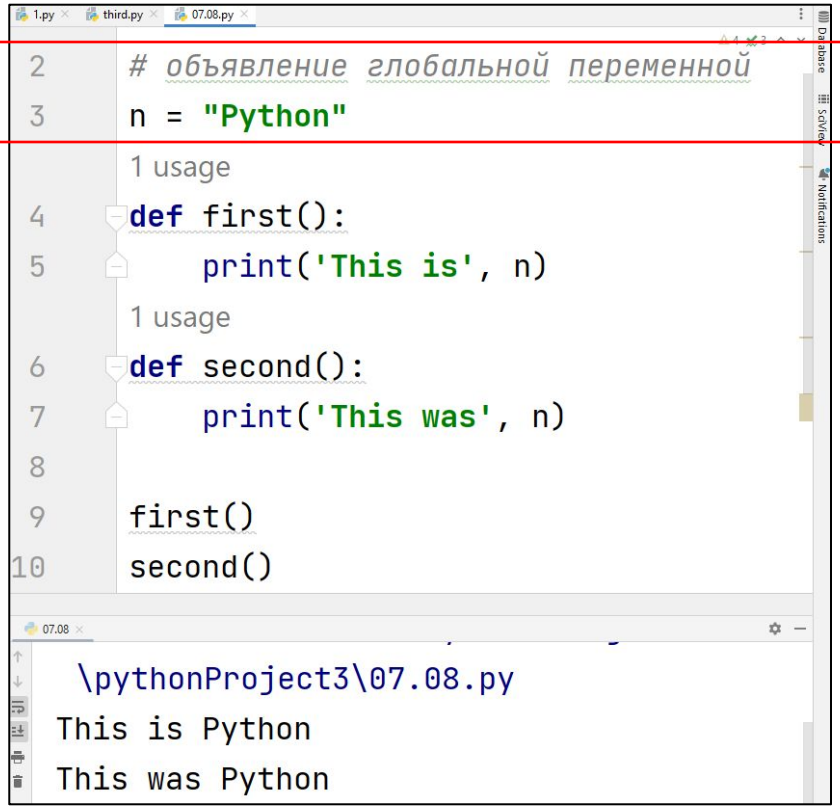
УЧЕБНЫЙ ЦЕНТР №1

Глобальные и локальные переменные

Знакомимся с новыми типами переменных

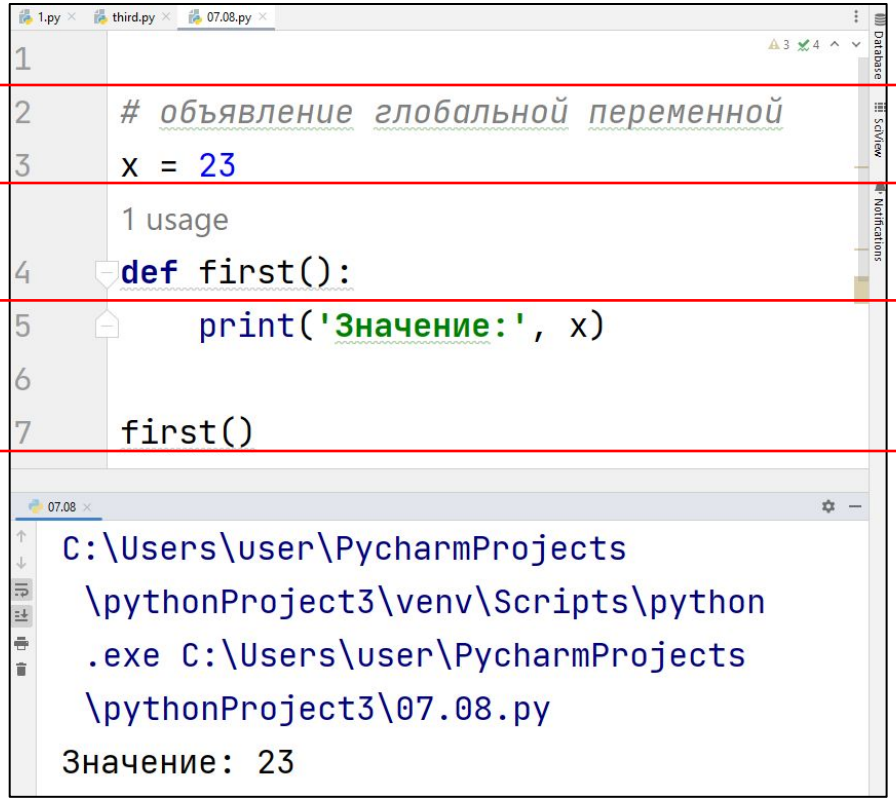


Немного теории:
глобальные и локальные переменные



Глобальные переменные могут быть объявлены вне функции, но использовать ей

Пример использования:
глобальные и локальные переменные



Здесь **глобальная переменная ‘x’** подставляется в функцию **‘first’**

Начинаем практику:
глобальные и локальные переменные

Задание.

Создайте программу, в которой есть две глобальные переменные и одна функция. В данной функции происходит сложение переменных и возвращение суммы.

Время на выполнение - 5 минут

Решение задания

1.py | third.py | 07.08.py

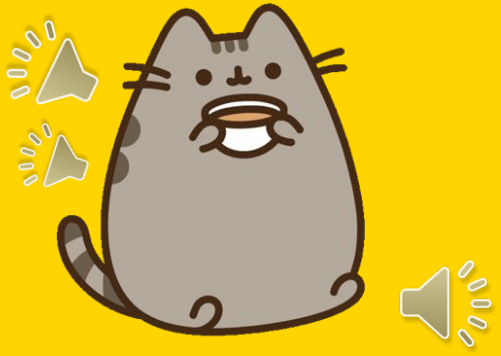
```
1  global a, b
2  a = 12
3  b = 15
   1 usage
4  def func():
5      print('Результат:', a+b)
6
7  func()
```

07.08

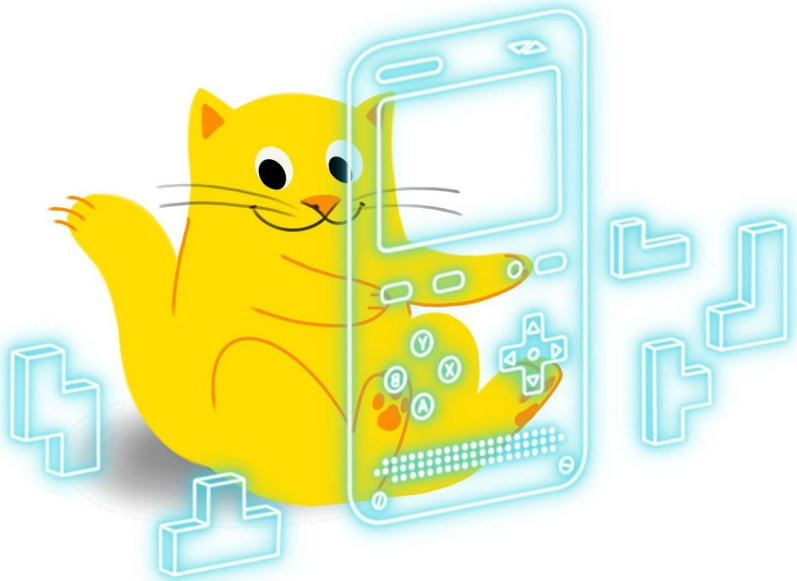
```
C:\Users\user\PycharmProjects
 \pythonProject3\venv\Scripts\python
 .exe C:\Users\user\PycharmProjects
 \pythonProject3\07.08.py
Результат: 27
```

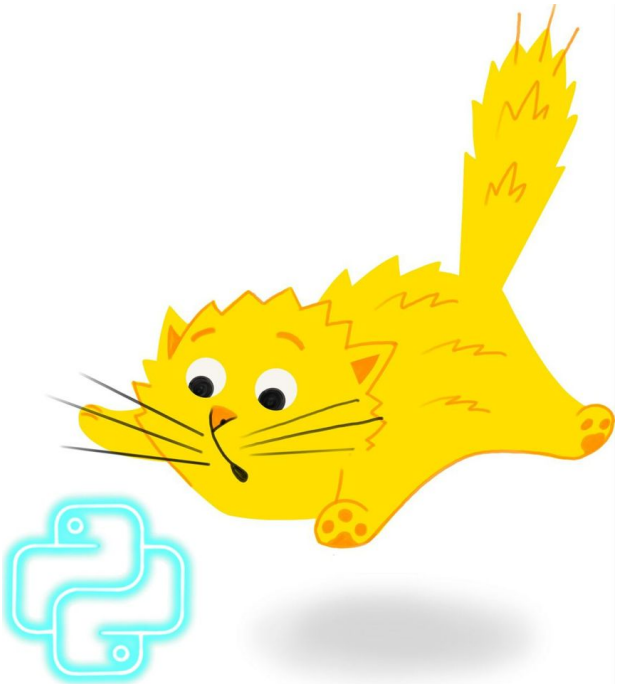
УЧЕБНЫЙ ЦЕНТР №1

Перерыв 10 минут



ПРАКТИЧЕСКИЕ ЗАДАЧИ





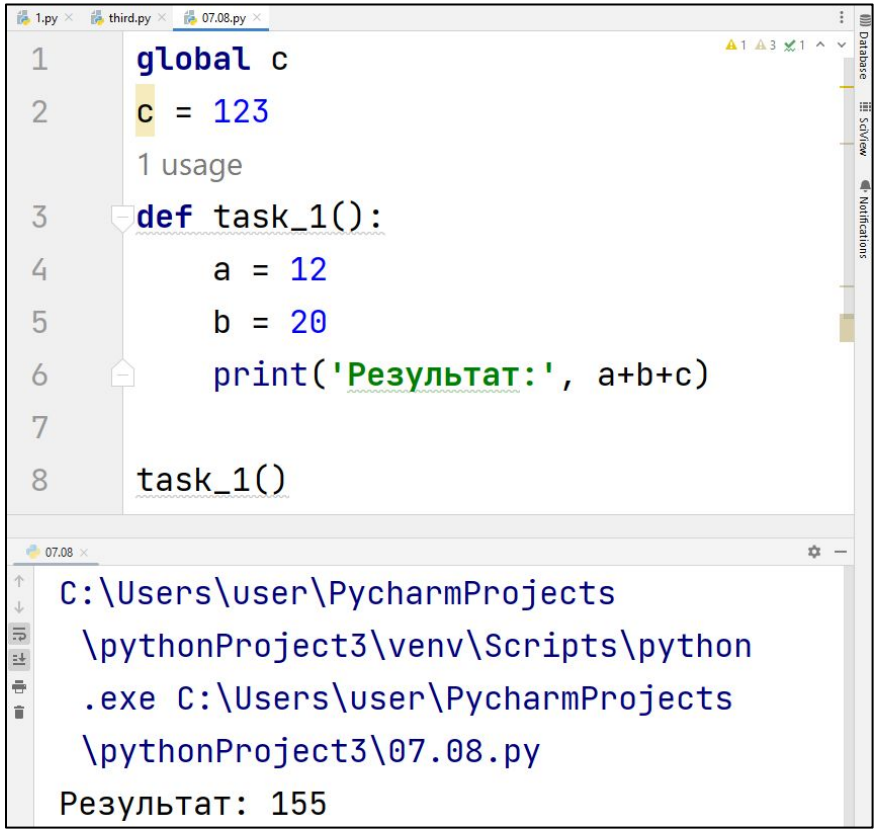
Практика. Задача №1

Написать программу с одной глобальной переменной **c**, и функцией, которая считает произведение трех целых чисел (включая переменную **c**).

Решение задачи №1

Напишем код для решения

и посмотрим на вывод:

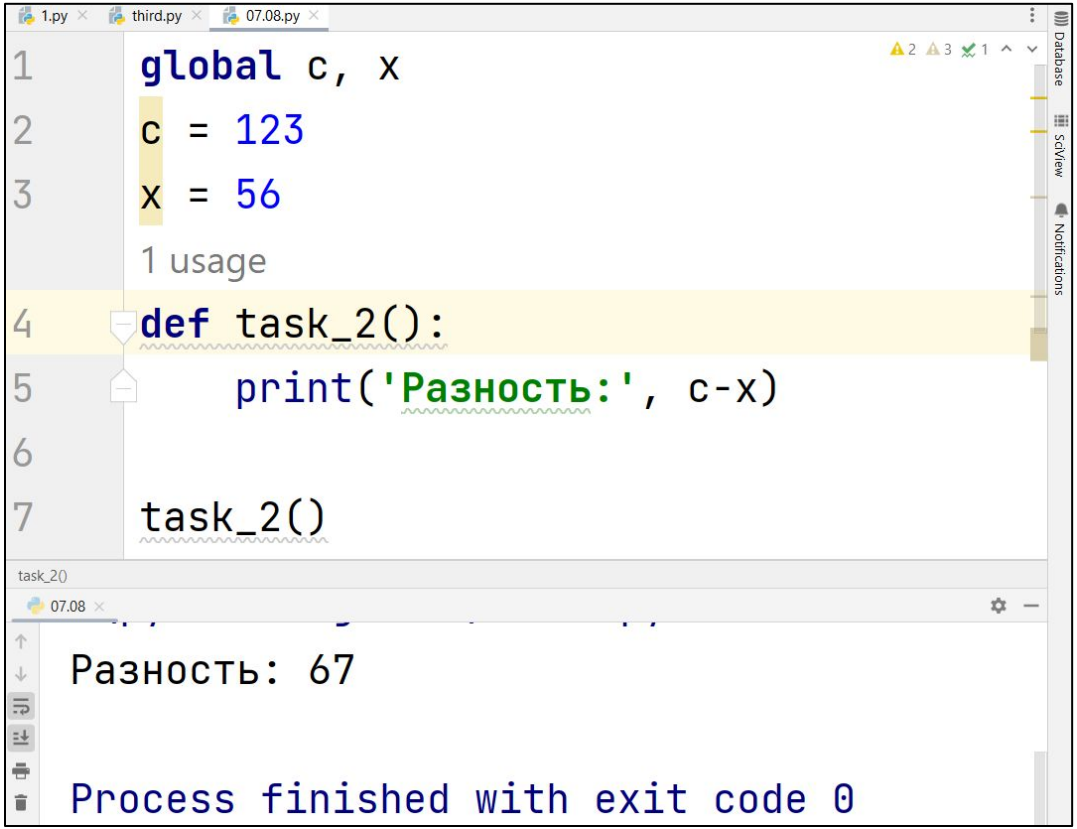


Практика. Задача №2

Написать программу с двумя глобальными переменными (имена задаются произвольно), которая вычисляет разность этих переменных.

Решение задачи №2

Напишем код для решения и посмотрим на вывод:

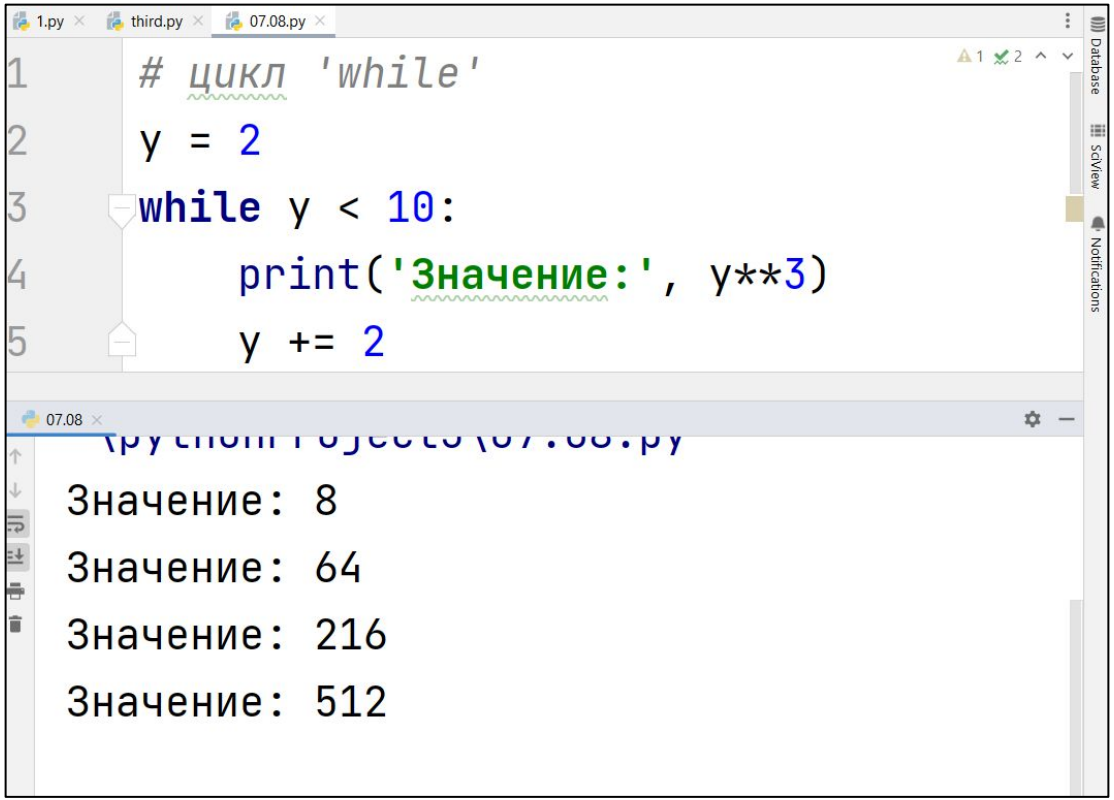


Практика. Задача №3

Создать программу с использованием цикла **while** и переменной **‘y’**, значение которой, изначально, равно 2. Данный цикл выводит значения переменной **‘y’** в третьей степени до тех пор, пока значение переменной **‘y’** не станет равным 10. При каждой итерации значение переменной **‘y’** увеличивается на 2.

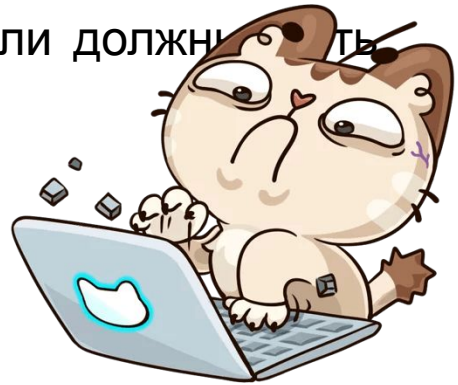
Решение задачи №3

Напишем код для решения и посмотрим на вывод:



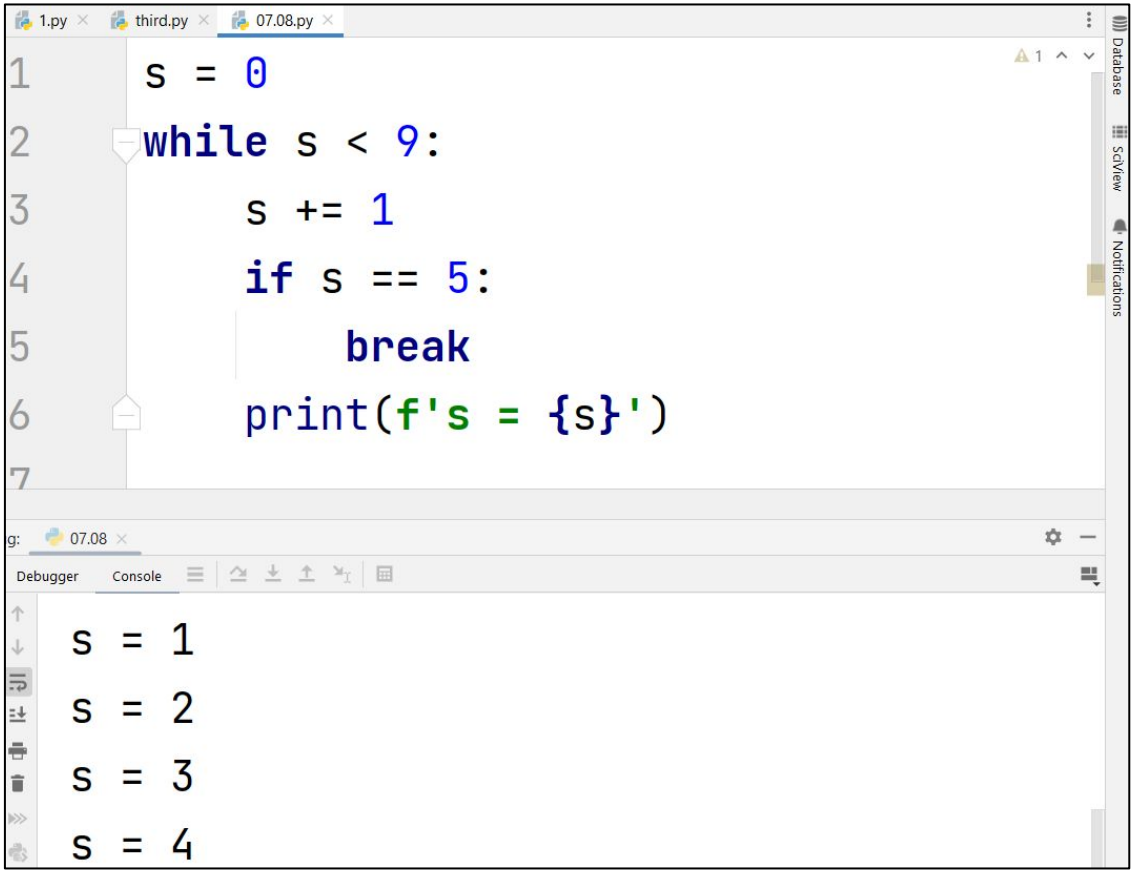
Практика. Задача №4

Создать программу, в которой реализован оператор выхода из цикла **break.** Изначально вводится переменная **‘s’** со значением **0**. При каждой итерации цикла значение переменной **‘s’** увеличивается на единицу. Когда значение переменной **‘s’** становится равным **5**, оператор **break** завершает работу цикла (на консоли должны быть выведены значения с **1** по **4** включительно).



Решение задачи №4

Напишем код для решения и посмотрим на вывод:



Практика. Задача №5

Создать программу с использованием цикла **for**. Данный цикл перебирает значения переменной **‘a’** от 3 до 7 и переменной **‘b’** от 10 до 13 и выводит соответствующие значения на консоль.

Решение задачи №5

Напишем код для решения и посмотрим на вывод:

1.py third.py 07.08.py

```
for a in range(3, 8):
    print('a =', a)
for b in range(10, 14):
    print('b =', b)
```

07.08

```
a = 3
a = 4
a = 5
a = 6
a = 7
b = 10
b = 11
b = 12
b = 13
```

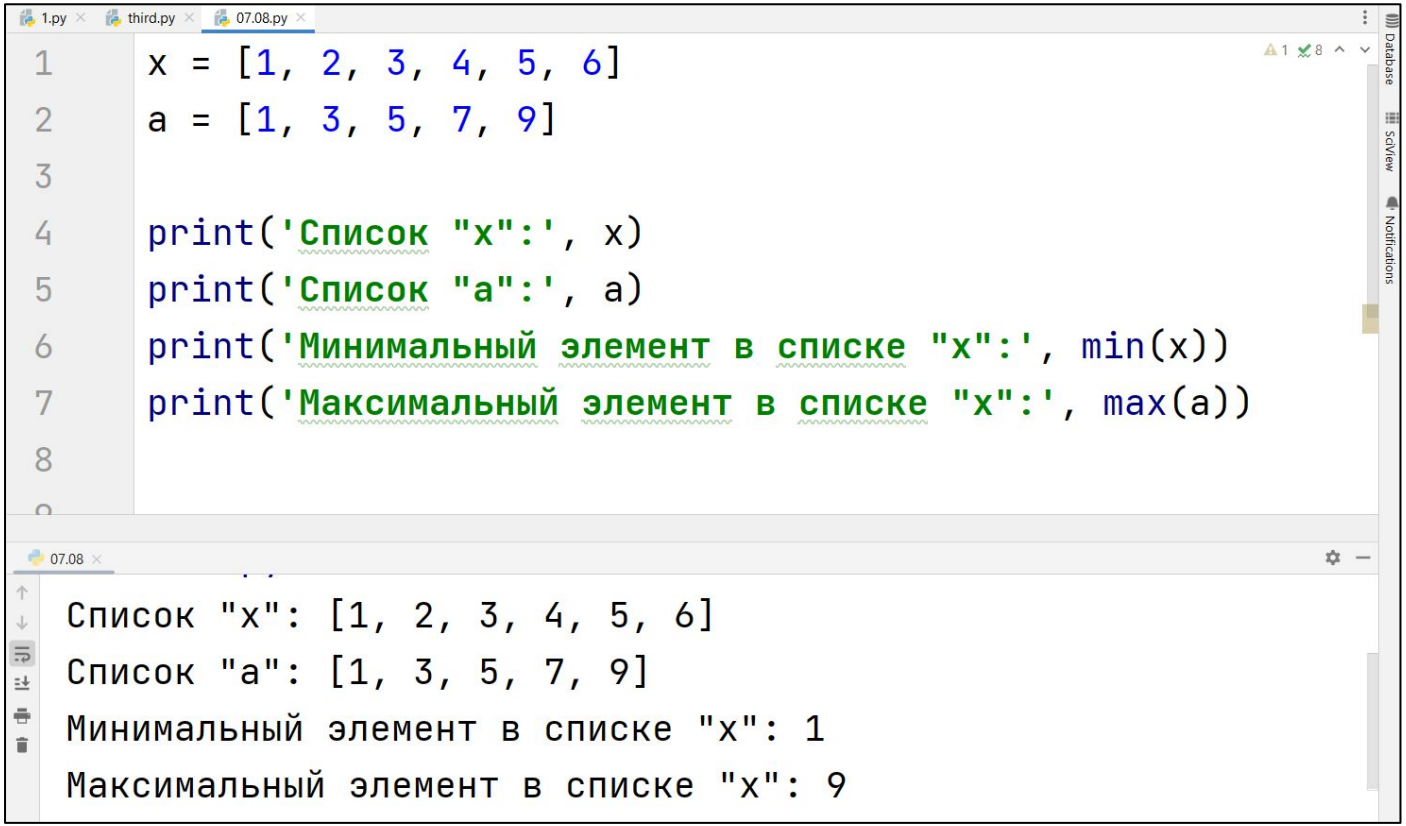
Практика. Задача №6

Создать программу, в которой вводятся списки **‘x’ = [1, 2, 3, 4, 5, 6]** и **‘a’ = [1, 3, 5, 7, 9]**. Необходимо найти максимальный элемент в списке **‘a’** и минимальный элемент в списке **‘x’**.

Данные значения вывести на консоль.

Решение задачи №6

Напишем код для решения и посмотрим на вывод:



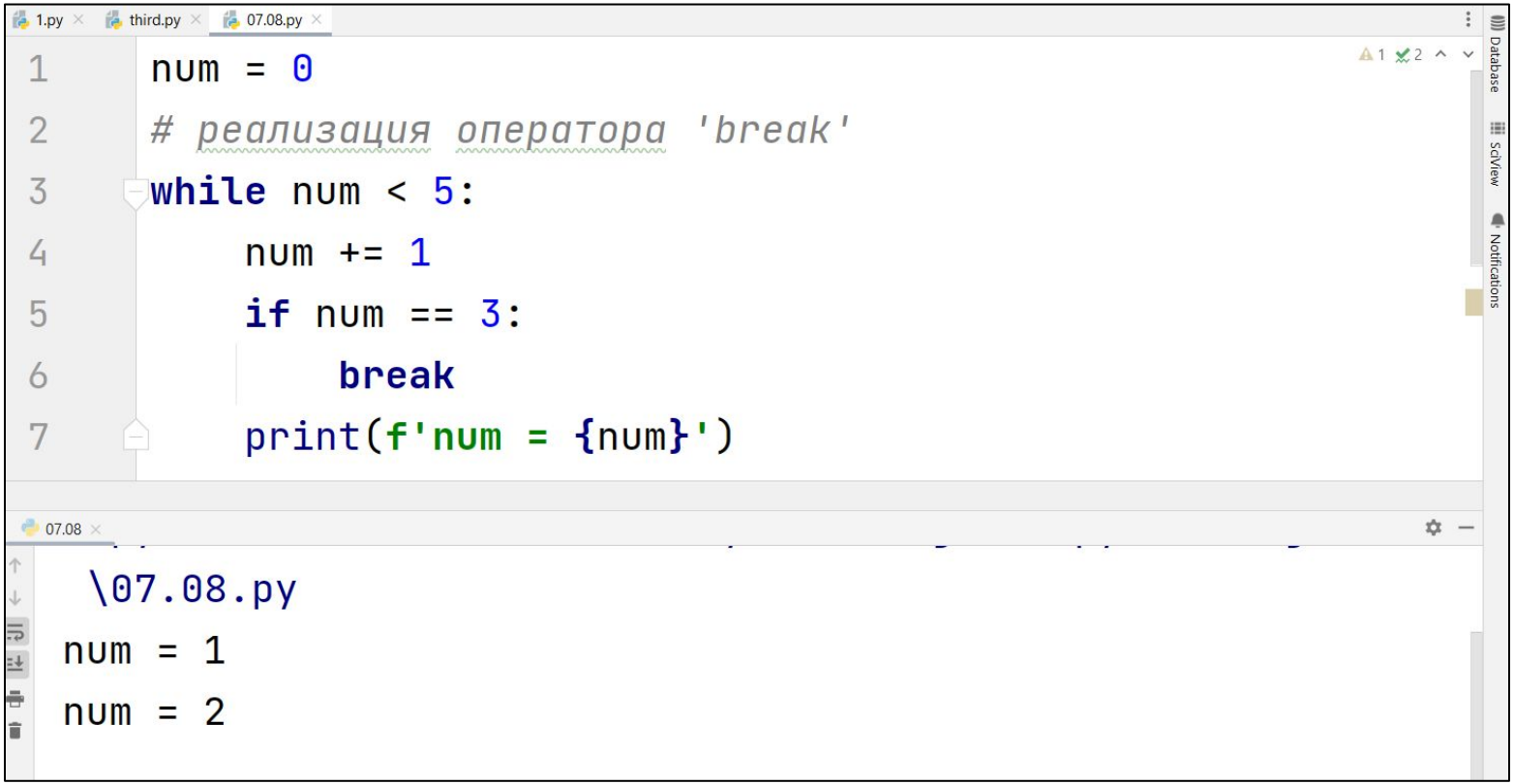
Практика. Задача №7

Создать программу, в которой реализован оператор выхода из цикла **break.** Изначально вводится переменная **‘num’** со значением **0**. При каждой итерации цикла значение переменной **‘num’** увеличивается на единицу. Когда значение переменной **‘num’** становится равным **3**, оператор **break** завершает работу цикла

(на консоли должны быть выведены оба значения: **1** и **2**).

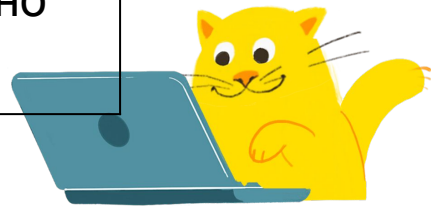
Решение задачи №7

Напишем код для решения и посмотрим на вывод:



Закрепление знаний

1. Какую роль выполняет ключевое слово **def** в языке *Python*?
2. Что такое **функция**?
3. Как можно передавать аргументы функции?
4. Какую роль выполняют функции в языке *Python*?
5. Какое максимальное количество аргументов можно передать функции на языке *Python* за один раз?



1С
УЧЕБНЫЙ ЦЕНТР №1

Спасибо за внимание!

До новых встреч!