АКАДЕМИЯ АЙТИ

День 12

academyit.ru

Понятие массива.

Типовые задачи с массивами: доступ к элементу, обход элементов, инициализация элементов

Турашова Анна Николаевна
Преподаватель
anna1turashova@gmail.com
Telegram: @anna1tur

Поверка домашнего задания

Задача 1

Пользователь ввёл некоторую комбинацию чисел.
Узнать, какой тип данный имеет то, что он ввёл.

Ввод: +4 Вывод: <class 'int'>	Ввод: 47.32 Вывод: <class 'float'>	Ввод: tom Вывод: <class 'str'>
Ввод: -123 Вывод: <class 'int'>	Ввод: -0.5697 Вывод: <class 'float'>	Ввод: 3,14 Вывод: <class 'str'>
Ввод: 0 Вывод: <class 'int'>	Ввод: 5e50 Вывод: <class 'float'>	Ввод: .50 Вывод: <class 'float'>

Задача 2

Пользователь ввёл строку с символами (через запятую с пробелом).
Узнать, какой тип данных имеет то, что он ввёл.

Ввод:
-123, 5.5, 5e50, tom, +4, .50, 3,14, -0.5697

Вывод:
<class 'int'>
<class 'float'>
<class 'float'>
<class 'str'>
<class 'int'>
<class 'float'>
<class 'str'>
<class 'float'>

Задача №3749. Количество различных чисел

Дан список чисел, который может содержать до 100000 чисел. Определите, сколько в нем встречается различных чисел.

Примечание. Эту задачу на Питоне можно решить в одну строчку.

Входные данные

Вводится список целых чисел. Все числа списка находятся на одной строке.

Выходные данные

Выведите ответ на задачу.

Примеры

входные данные
1 2 3 2 1
выходные данные
3

входные данные
1 2 3 4 5 6 7 8 9 10
выходные данные
10

Задача №3751. Пересечение множеств

Даны два списка чисел, которые могут содержать до 10000 чисел каждый. Выведите все числа, которые входят как в первый, так и во второй список в порядке возрастания.

Примечание. И даже эту задачу на Питоне можно решить в одну строчку.

Входные данные

Вводятся два списка целых чисел. Все числа каждого списка находятся на отдельной строке.

Выходные данные

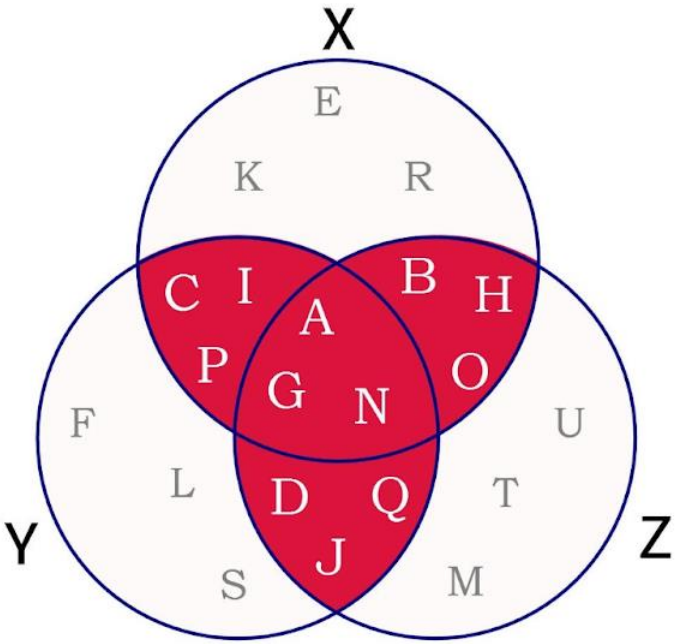
Выведите ответ на задачу.

Примеры

входные данные
1 3 2 4 3 2
выходные данные
2 3

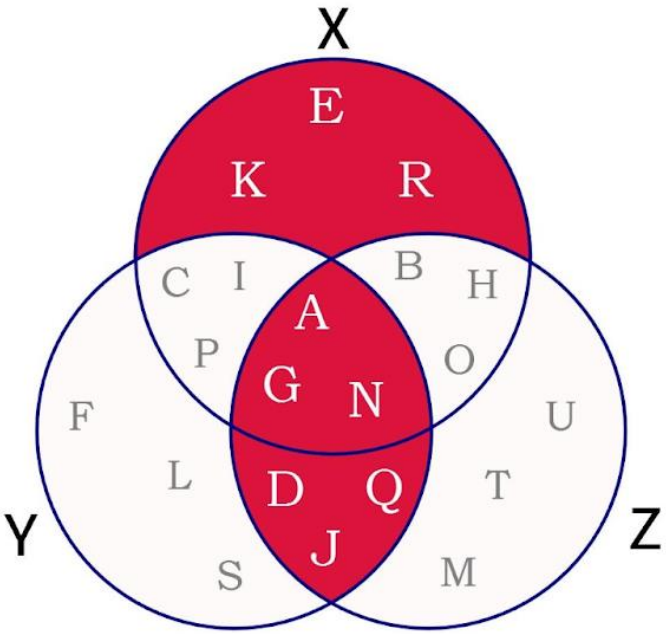
Задача множ-4

Даны три следующих множества:
X = {'A', 'P', 'I', 'B', 'E', 'H', 'C', 'G', 'O', 'R', 'K', 'N'}
Y = {'A', 'J', 'S', 'P', 'I', 'L', 'D', 'C', 'G', 'Q', 'F', 'N'}
Z = {'A', 'J', 'N', 'B', 'U', 'T', 'H', 'D', 'G', 'Q', 'O', 'M'}
Попробуйте подобрать формулу для:



Задача множ-5

Даны три следующих множества:
X = {'A', 'P', 'I', 'B', 'E', 'H', 'C', 'G', 'O', 'R', 'K', 'N'}
Y = {'A', 'J', 'S', 'P', 'I', 'L', 'D', 'C', 'G', 'Q', 'F', 'N'}
Z = {'A', 'J', 'N', 'B', 'U', 'T', 'H', 'D', 'G', 'Q', 'O', 'M'}
Попробуйте подобрать формулу для:



Функции

Понятие функции

Функция — это фрагмент кода, который выполняет какую-то задачу и к нему можно обратиться по имени из разных частей программы. Функция может получить данные на входе, выполнить инструкции (команды) и вернуть новые данные на выходе, если указать параметры функции и вернуть результат.

Функции должны делать одну вещь и делать ее хорошо и делать только ее. Роберт С. Мартин

Синтаксис функции

```
def имя_функции(параметры):
    определение_функции
```

Ключевое слово **def** сообщает Python, что вы определяете функцию. После def вы указываете имя функции; оно должно отвечать тем же правилам, что и имена переменных. Как правило, имя функции отвечает на вопрос: **Что сделать?**

Параметры - переменные в круглых скобках после имени функции; они получают значения при вызове функции

Определение функции - содержит исполняемый код функции. Любой код с отступом в четыре пробела после двоеточия является определением функции.

Вызов функции - обращение к функции по имени с передачей функции входных данных, необходимых для выполнения указаний и возвращения вывода.

Зачем нужны функции?

- Функция – вспомогательный алгоритм, который может быть вызван из любого места программы, в том числе из других функций
- Применение функций
  - Разбиение программы на логические части
  - Повторное использование кода
  - Создание новых действия и операций
  - Сокрытие деталей реализации

Функции

- Функция в Python может возвращать значение, которое используется в выражениях
- По умолчанию функция возвращает пустое значение
- Перед использованием функция должна быть определена с помощью def

define – определить

```
def <имя функции> ([<список параметров>]):
    <тело функции>
```

Пример функции

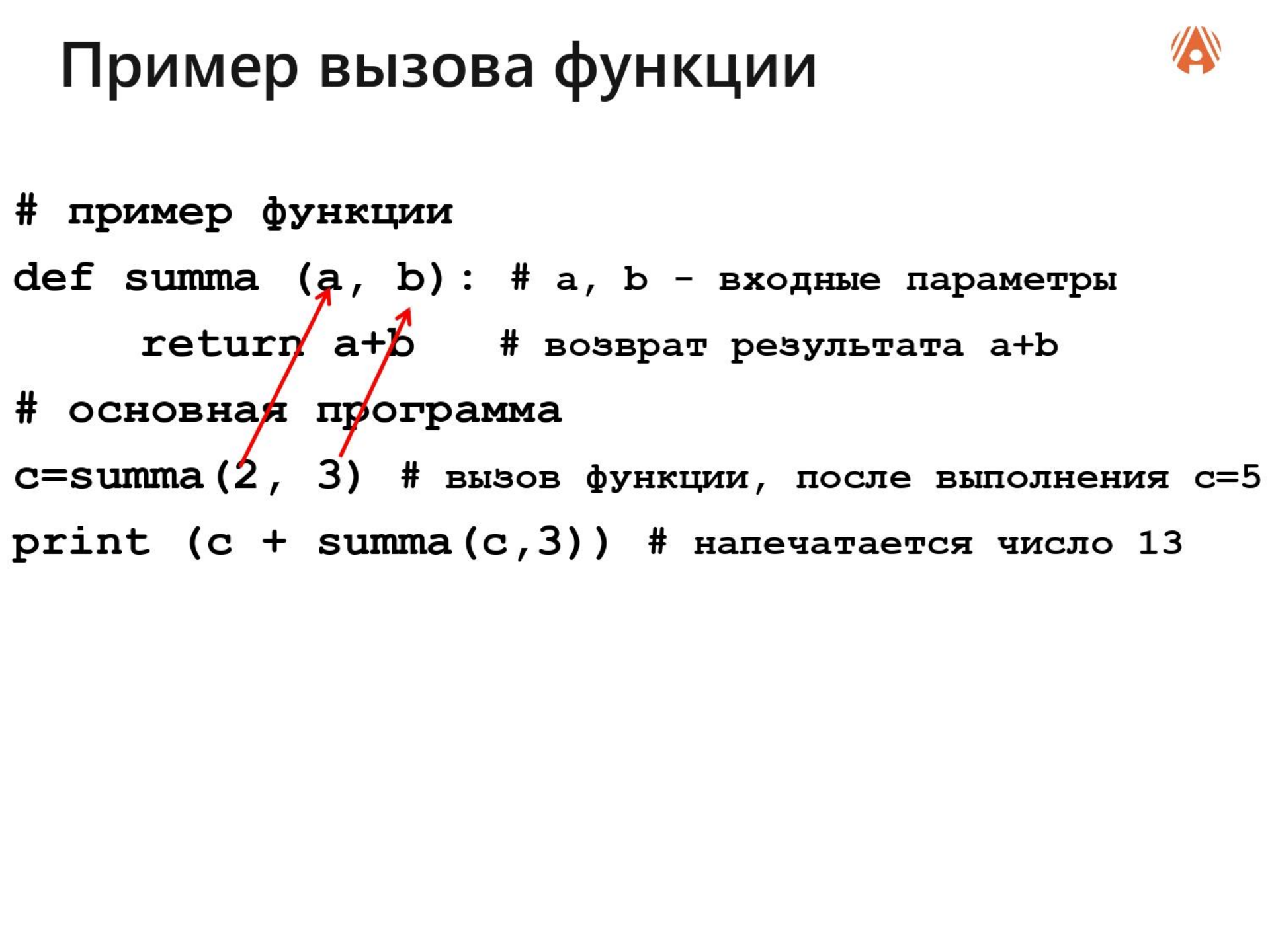
```
def add(x, y):
    return x + y
```

- Вызов на выполнение функции осуществляется по ее имени, а также с указание списка фактических параметров функции (аргументов)
- Параметры могут отсутствовать, тогда скобки будут пустые
- Инструкция return говорит, что нужно вернуть значение из функции в точку ее вызова.
- В функции может быть несколько операторов return, после выполнения любого из них работа функции заканчивается
- Функция может быть любой сложности и возвращать любые объекты (списки, кортежи, и даже функции!)

Примеры функции

```
def say_hi():                # функция без парамтров
    print("Hello bro!")

def show_info(name, age):    # функция с параметрами
    print("Name:", name)
    print("Age:", age)

def area(a, b):              # функция с параметрами
    return a * b             # и возвращаемым значением
```

Пример вызова функции



```
# пример функции
def summa (a, b): # a, b - входные параметры
    return a+b    # возврат результата a+b
# основная программа
c=summa(2, 3) # вызов функции, после выполнения c=5
print (c + summa(c,3)) # напечатается число 13
```

Возврат нескольких значений

В Python позволительно возвращать из функции несколько объектов, перечислив их через запятую после команды **return**:

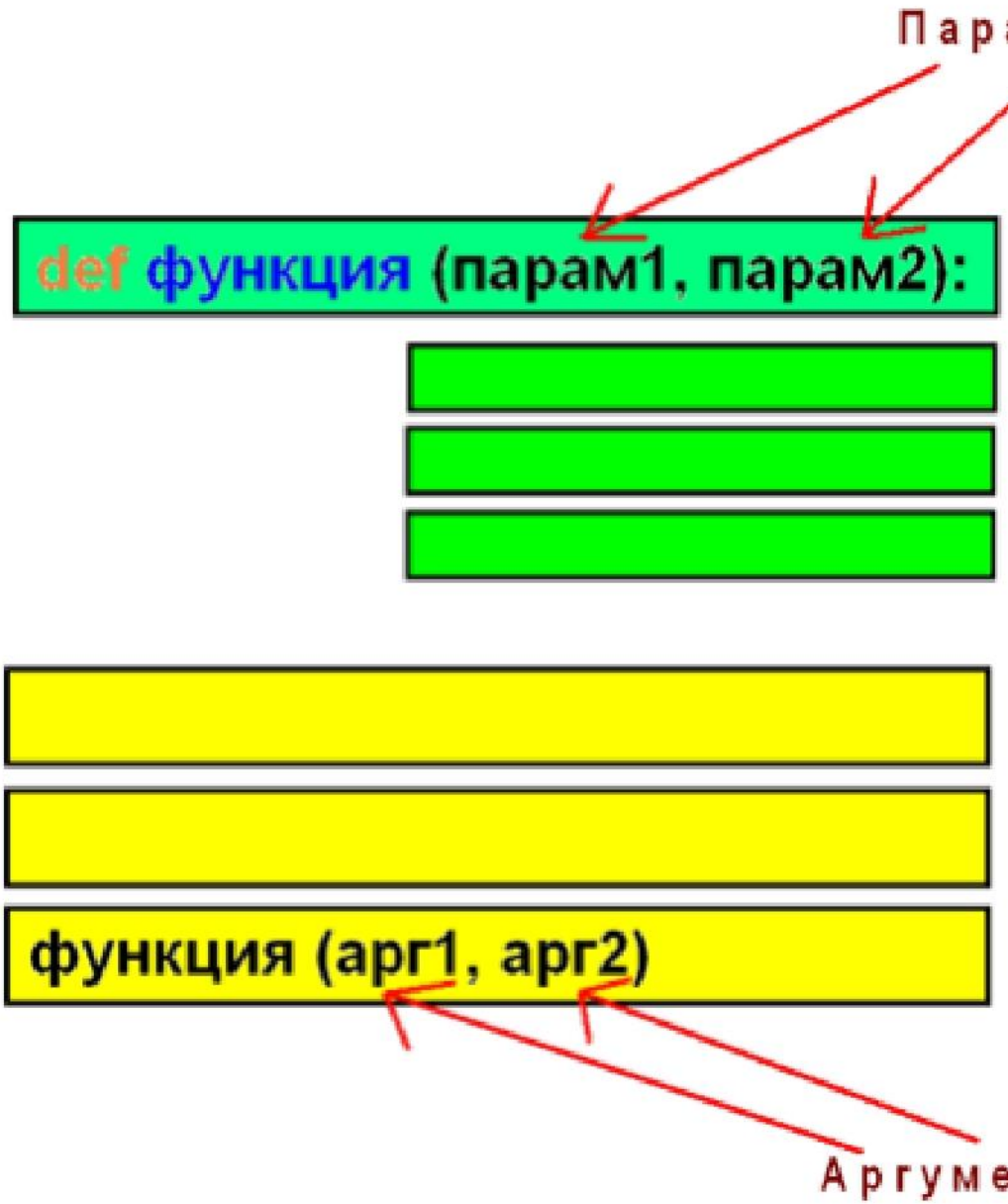
```
def rectangle(a, b):
  perimeter = 2 * (a + b)
  square = a * b
  return perimeter, square

p, s = rectangle(2, 5)
```

Перечисление значений через запятую в операторе return создает объект типа **tuple** (кортеж).

```
print(rectangle(2, 5))
```

Результат: (14, 10)

Параметры и аргументы функции



Пример:

```
>>> def mathem (a, b):
        a = a / 2
        b = b + 10
        print (a * b)


>>> num1 = 100
>>> num2 = 12
>>> mathem (num1, num2)
1100.0
```

Задачи

1. Напишите функцию, которая получает на входе: имя, возраст и телефон. Далее выводит информацию о человеке:

```
Name: Alex
Age: 25
Phone: 7934534122
```

2. Напишите функцию, которая получает на входе радиус, находит площадь круга и возвращает ее в качестве результата.

3. Напишите функцию генерации случайного пароля длиной 6 символов состоящего из чисел и символов латинского алфавита. Затем измените функцию, чтобы она генерировала пароль любой заданной длины.

5. Напишите функцию возведения числа в степень

Алгоритм решения задачи:

Чтобы возвести число в степень, его надо умножить само на себя n количество раз, равное показателю степени. Т.е. возведение числа x в степень n будет выглядеть так:

x**n = x1 * x2 * x3 * ... * xn

Пример:

2**4 = 2 * 2 * 2 * 2 = 16

Если n равно нулю, то, какое бы число не стояло в основании степени, результат всегда будет равен единице:

x**0 = 1.

Если показатель степени отрицателен (n < 0), то результат определяется такой формулой:

x**n = 1 / (x1 * x2 * x3 * ... * xn).

Задачи

1. Напишите функцию, которая принимает число в качестве ввода, возводит его в квадрат и возвращает.

2. Напишите функцию, которая на входе получает имя человека и всегда возвращает имя с заглавной буквы

3. Напишите функцию которая проверяет является ли число четным. На входе функция получает целое число и если оно четное, то вернуть True, а если нечетное, то вернуть False

4. Напишите функцию вычисления факториала. Факториалом числа называют произведение всех натуральных чисел до этого числа включительно. Например, факториал числа 5 равен 1*2*3*4*5 = 120. Записывается факториал так: 5! = 120.

Алгоритм вычисления факториала:

1. Получить от пользователя число, для которого надо найти факториал

2. Создать переменную, накапливающую произведение натуральных чисел и присвоить начальное значение 1

3. Присвоить переменной-счетчику значение 2

4. Пока счетчик не достигнет числа, введенного пользователем:

    1. умножить значение переменной, в которой накапливается произведение, на значение переменной счетчика

    2. увеличить счетчик на 1

6. Напишите функцию sum_range(start, end), которая суммирует все целые числа от значения «start» до величины «end» включительно. Если пользователь задаст первое число большее чем второе, просто поменяйте их местами.
6. Создайте функцию three_args(), которая принимает 1, 2 или 3 строго ключевых параметра. В результате ее работы на печать в консоль выводятся значения переданных переменных, но только если они не равны **None**. Получим, например, следующее сообщение: «Переданы аргументы: var1 = 2, var3 = 10».
6. Чтобы проверить понимание параметров и область их видимости, Николай создал 3 функции (представлены ниже). Попытайтесь предугадать, как поведет себя каждая из них при запуске (возникнут ли ошибки, что возвратится).

```
def func1():
    param = 4

    def inner():
        param += 1

    return param
```

```
def func2():
    param = 4

    def inner(var):
        var += 1

    inner(param)
    return param
```

```
def func3():
    param = 4

    def inner(var):
        var += 1
        return var

    param = inner(param)
    return param
```

Функции: дополнительные возможности

Значения по умолчанию

В Python у функций бывают параметры, которым уже присвоено значение ***по умолчанию***.

В таком случае, при вызове можно не передавать соответствующие этим параметрам аргументы. Хотя можно и передать. Тогда значение по умолчанию заменится на переданное.

```
def sum3(a,  b = 0, c = 0):
    return a + b + c

print(sum3(2))
print(sum3(2, 4))
print(sum3(2, 4, 10))
```

Явное указание аргументов

```
def rectangle(a, b):
  perimeter = 2 * (a + b)
  square = a * b
  return perimeter, square

print(rectangle(b = 2, a = 5))
```

Произвольное количество аргументов

Если поставить * перед именем параметра, это имя будет принимать не один аргумент, а несколько.

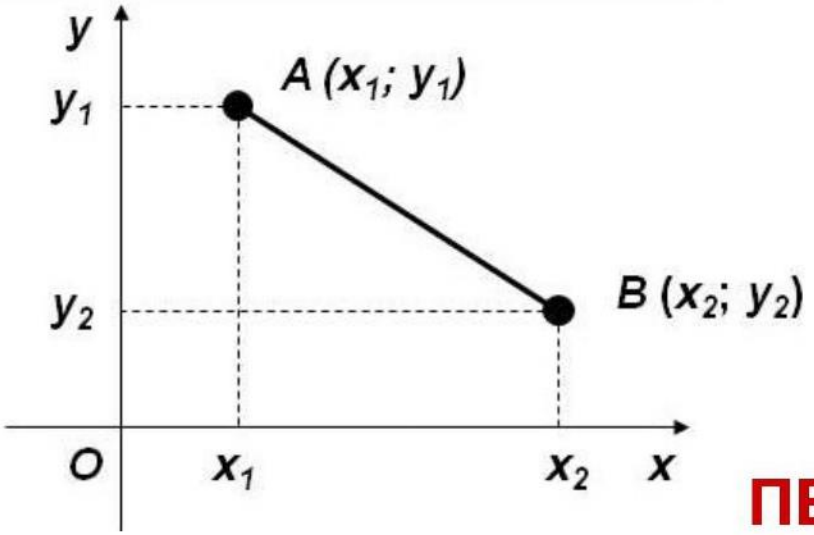
Аргументы передаются как **кортеж** и доступны внутри функции под тем же именем, что и имя параметра, только без *.

Например:

```
def summa(*nums):
  sum = 0
  for n in nums:
     sum += n
  return sum

print(summa(1, 2, 3, 5, 6))
```

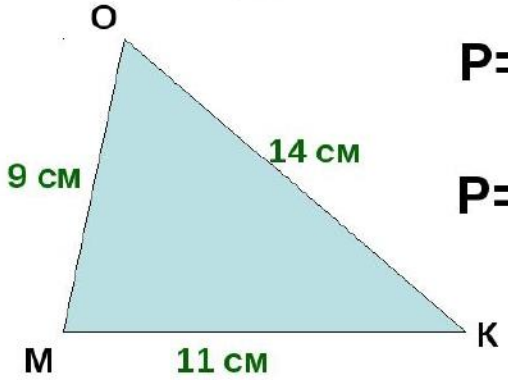
Пример. Периметр треугольника.

$$d(A,B)=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$



ПЕРИМЕТР ТРЕУГОЛЬНИКА

СУММА ДЛИН ВСЕХ ЕГО СТОРОН



P= MO+ OK+ KM

P= 9+ 14+ 11 =34(см)

Домашнее задание

Задача 1

Пользователь вводит какие-то символы. Написать три функции:

isint – проверяет, можно ли число без ошибок преобразовать в int
isfloat – проверяет, можно ли число без ошибок преобразовать в float

По желанию:
Можете так же создать islist, istuple, isset, isdict

```
def isint(s):
    if s == ?:
        return True
    else:
        return False
s = input()
print(isint(s))
```

```
def isfloat(s):
    if s == ?:
        return True
    else:
        return False
s = input()
print(isfloat(s))
```

Задача 2

Напишите "функцию голосования"

def Election(x, y, z)

возвращающую то значение (true или false), которое среди значений ее аргументов x, y, z встречается чаще.

Входные данные

Вводится 3 числа - *x*, *y* и *z* (*x*, *y* и *z* равны `0` или `1`, `0` соответствует значению `false`, `1` соответствует значению `true`).

Выходные данные

Необходимо вывести значение функции от *x*, *y* и *z*.

Примеры

входные данные
`0 0 1`
выходные данные
`0`

Задача 3

Напишите функцию, которая создаёт комбинацию двух списков таким образом:

```
[1, 2, 3] (+) [11, 22, 33] -> [1, 11, 2, 22, 3, 33]
```

Задача 4

У вас есть список чисел, напиши функцию, которая составляет из этих чисел максимальное число. Например:

```
[61, 228, 9] -> 961228
```

Задача 5

У вас есть девять цифр: 1, 2, ..., 9. Именно в таком порядке.
Вы можете вставлять между ними знаки «+», «-» или ничего.
У вас будут получаться выражения вида 123+45-6+7+89.
Найдите все из них, которые равны 100.

АПЛАНА
ГРУППА КОМПАНИЙ

Входит в ГК Аплана

АКАДЕМИЯ АЙТИ

Основана в 1995 г.

E-learning и очное обучение

Направления обучения:

- Информационные технологии
- Информационная безопасность
- ИТ-менеджмент и управление проектами
- Разработка и тестирование ПО
- Гос. и муниципальное управление

Филиалы:

Санкт-Петербург, Казань, Уфа, Челябинск, Хабаровск, Красноярск, Тюмень, Нижний Новгород, Краснодар, Волгоград, Ростов-на-Дону

Головной офис в Москве

Разработка программного обеспечения и информационных систем

Ежегодные награды Microsoft, Huawei, Cisco и другие

Программы по импортозамещению

Ресурсы более 400 высококлассных экспертов и преподавателей

Сеть региональных учебных центров по всей России

Крупные заказчики

СБЕР

РЖД

Билайн

TATNEFT

НОРНИКЕЛЬ

100+
сотрудников

АКАДЕМИЯ АЙТИ

Спасибо за внимание!

Центральный офис:
Москва, Варшавское шоссе 47, корп. 4, 7 этаж
Тел: +7 (495) 150-96-00
academy@it.ru
academyit.ru