

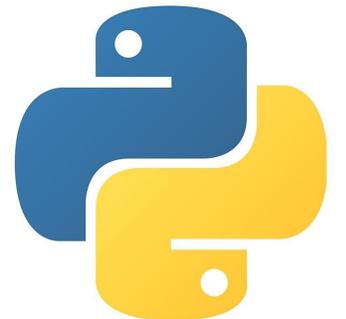


python

Занятие 2. Условный оператор

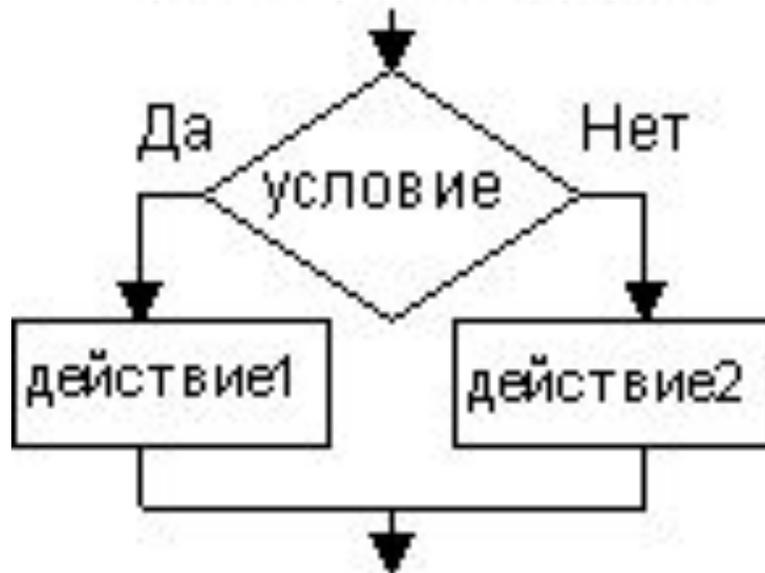
Ветвление

Ветвление алгоритма – способ организации порядка выполнения действий, когда в зависимости от **выполнения** или **невыполнения** некоторого условия выполняется одна или другая последовательность шагов.

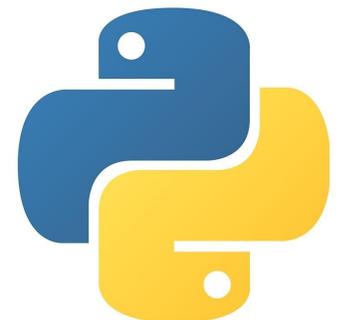


Формы ветвлений

Полная форма



Неполная форма



Как «разветвить» программу?



Условный оператор

Условный оператор позволяет организовать **ветвление алгоритма**

Неполная форма:

if *условие:*

действия, выполняемые при истинности условия



Условный оператор

Полная форма:

if *условие:*

действия, выполняемые при истинности условия

else:

действия, выполняемые если условие ложно

Tab



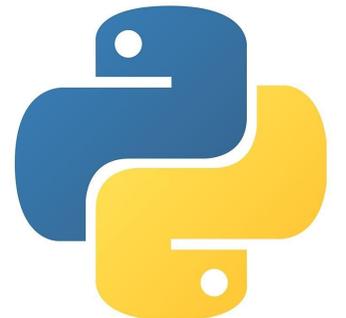
Как должно выглядеть условие?



Составление условий

Условие оператора `if` должно иметь логический тип. Иными словами, перед двоеточием должно стоять некое выражение, в результате которого получается значение типа `bool` (истина или ложь)

ВАЖНО: блок кода, который необходимо выполнить, в случае истинности выражения, отделяется четырьмя пробелами слева или клавишей `Tab`



Составление условий

Пример:

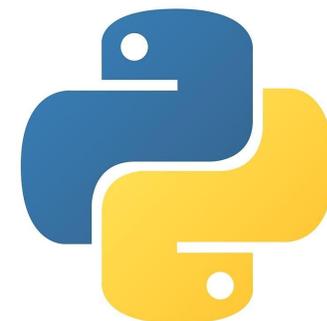
```
t=True
if t:
    print("True")
else:
    print("False")
```



Логические операторы

Допустим, у нас есть два числа a и b . К ним можно применить следующие **операторы сравнения**

Оператор	Смысл
$a > b$	a больше b
$a < b$	a меньше b
$a \geq b$	a больше или равен b
$a \leq b$	a меньше или равен b
$a == b$	a равен b
$a != b$	a не равен b

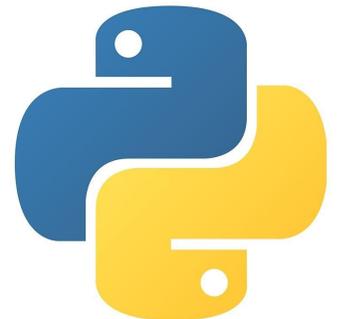


Логические связки

Логические связки нужны для построения сложных условий.

Три основные логические связки:

- Логическое «не»
- Логическое «и»
- Логическое «или»

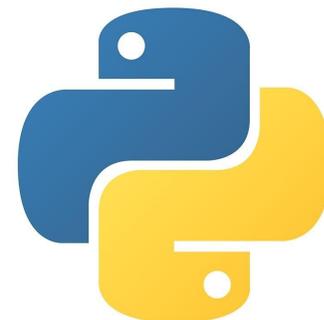


Логическое «не»

Логическое «не» (отрицание, инверсия) изменяет значение выражения на противоположное.

Отрицание обозначается с помощью слова «**not**»

```
t=False
if not t:
    print("t - ложно")
else:
    print("t - истина")
```



Логическое «и»

Логическое «и» (конъюнкция) истинно тогда, когда истинны оба аргумента.

Обозначается с помощью служебного слова «**and**»

```
k=6
```

```
m=8
```

```
if k>5 and m>7:
```

```
    print("оба условия выполнены")
```

```
else:
```

```
    print("как минимум одно не выполнилось")
```



Логическое «или»

Логическое «или» (дизъюнкция) истинно тогда, когда истинным является хотя бы один аргумент.

Обозначается с помощью слова «**or**»

```
k=6
```

```
m=-8
```

```
if k>0 or m>0:
```

```
    print("хотя бы одно число положительное")
```

```
else:
```

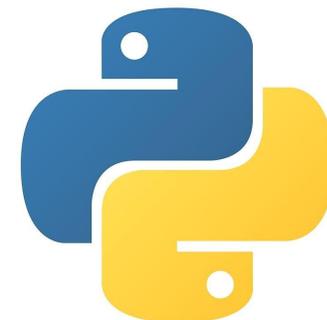
```
    print("все числа отрицательны или нули")
```



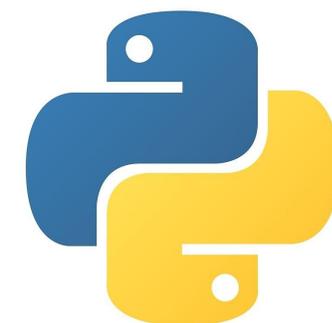
Цепочки if – elif - else

Можно составлять сложные последовательности проверок, например:

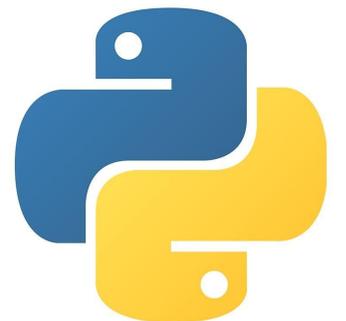
```
x=int(input())
if x>=0 and x<=9:
    print("Однозначное")
elif x>9 and x<=99:
    print("Двузначное")
elif x>99 and x<1000:
    print("Трёхзначное")
else
    print(" Многочисленное")
```



Задачи!



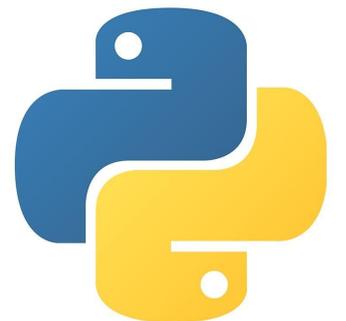
Вводится целое число.
Проверить, является ли оно четным



Вводится число.
Вывести «+», если оно положительное,
«-», если оно отрицательное
и «ноль», если число равно 0.
Использовать if-elif-else



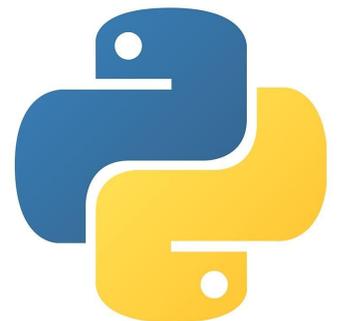
Даны два числа. Найти и вывести
большее из них



Вводятся длины сторон треугольника.
Проверить, существует ли треугольник
с такими сторонами



Вводятся длины сторон треугольника.
Проверить, существует ли треугольник
с такими сторонами



Написать программу, решающую квадратное уравнение ($ax^2+bx+c=0$). Вводятся a, b, c .



Вводится номер билетика автобуса. Программа должна определить, можно ли его назвать «счастливым»



На сегодня достаточно



Жду вас на следующих занятиях!
На забывайте тетрадь и ручку!

