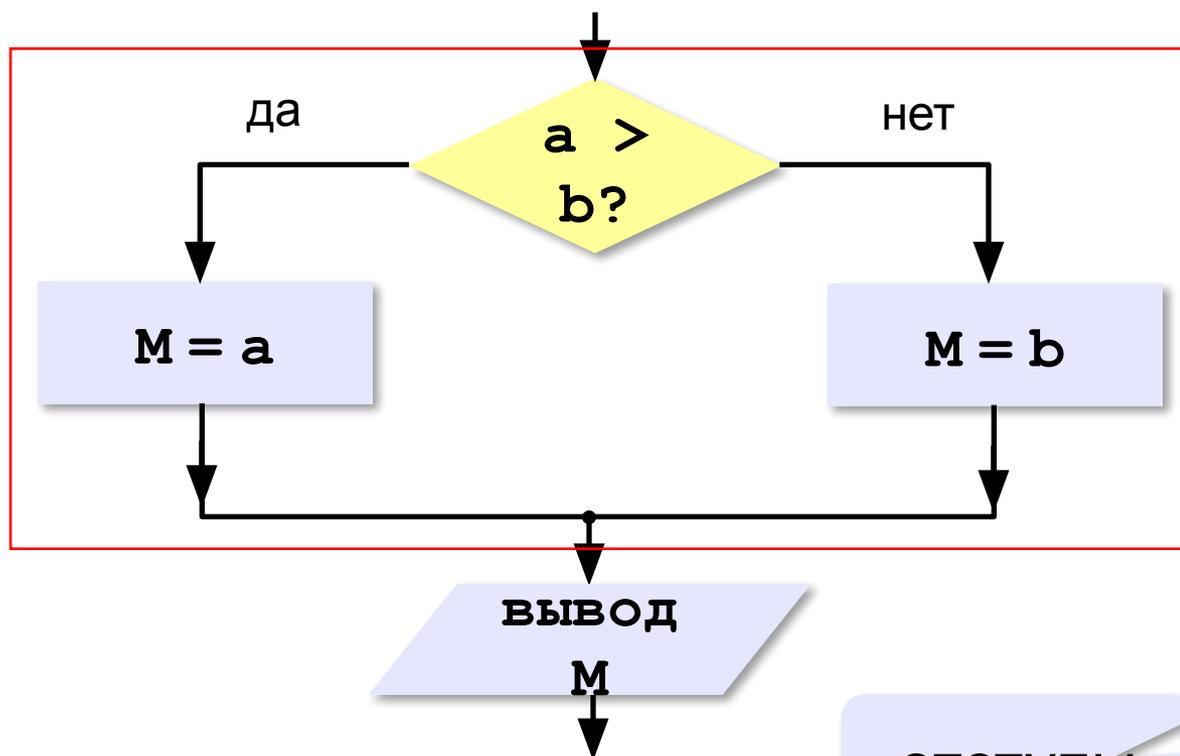


# Организация ветвления в Python

# Условный оператор

Задача: **изменить порядок действий** в зависимости от выполнения некоторого условия.



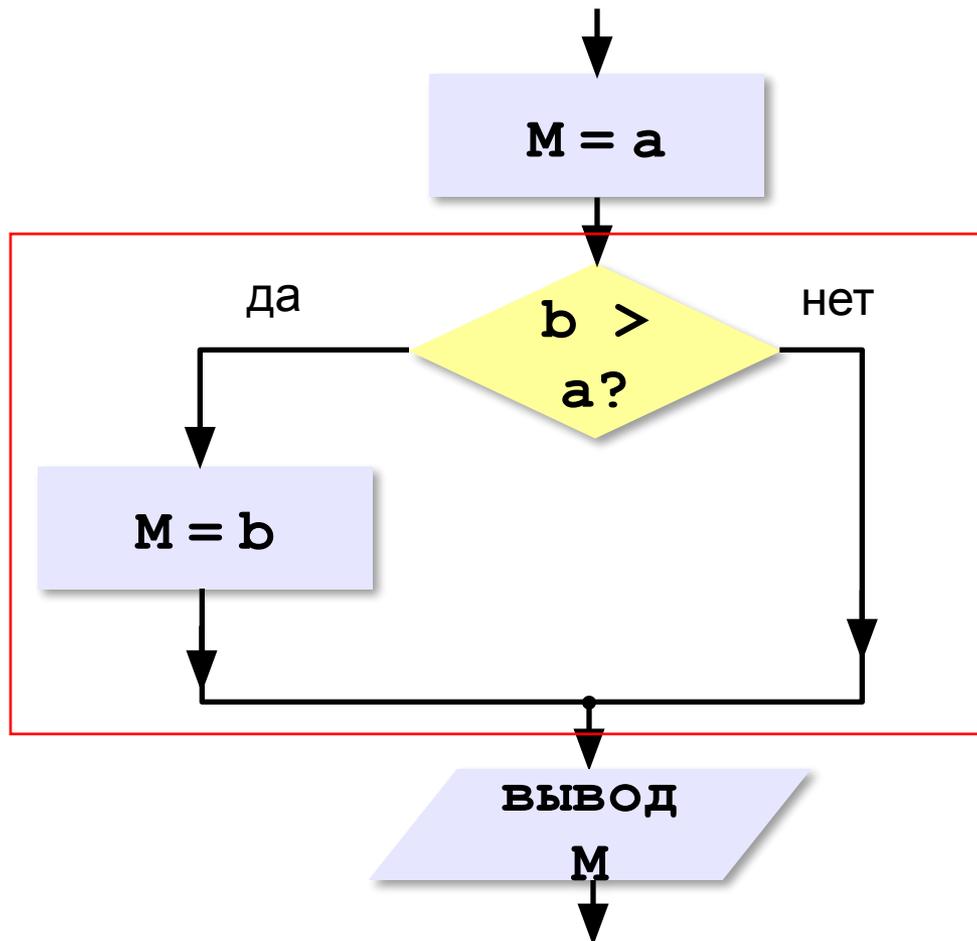
полная  
форма  
ветвления

? Если  $a = b$ ?

```
if a > b:  
    M = a  
else:  
    M = b
```

отступы

# Условный оператор: неполная форма



```
M = a
if b > a:
    M = b
```

неполная  
форма  
ветвления

Решение в стиле Python:

```
M = max(a, b)
```

```
M = a if a > b else b
```

# Условный оператор

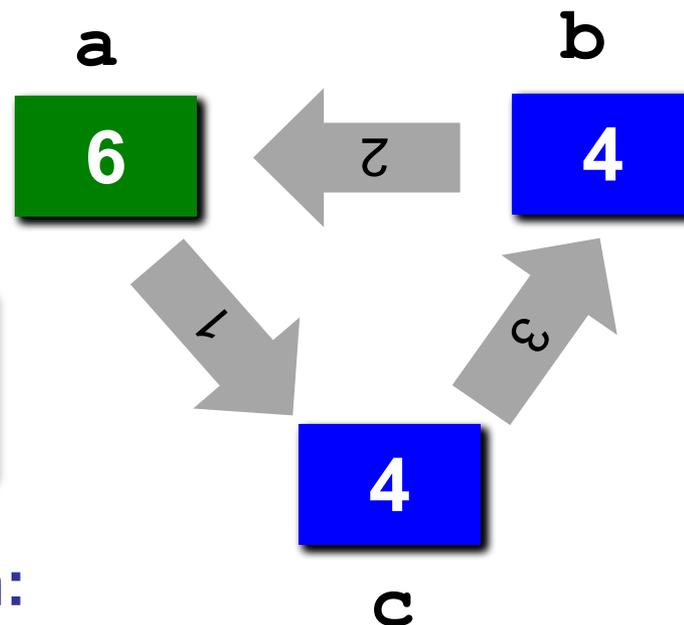
```
if a < b:
```

```
    c = a
```

```
    a = b
```

```
    b = c
```

? Что делает?



? Можно ли обойтись без переменной **c**?

Решение в стиле Python:

```
a, b = b, a
```

## Составить программу для решения уравнения

---

$$ax^2 + bx + c = 0$$

### Формат входных данных:

значения коэффициентов  $a$ ,  $b$  и  $c$  вводятся с клавиатуры через пробел в одной строке

## Программа:

```
from math import sqrt
print("Введите a, b, c:")
a, b, c = map(float, input().split())
D = b*b - 4*a*c
if D < 0:
    print("Нет")
else:
    x1 = (-b + sqrt(D)) / (2*a)
    x2 = (-b - sqrt(D)) / (2*a)
    print("x1={:5.3f} x2={:5.3f}".format(
        x1, x2))
```



Что делает?

# Знаки отношений

---

- >** **<** больше, меньше
- >=** больше или равно
- <=** меньше или равно
- ==** равно
- !=** не равно

# Вложенные условные операторы

Задача: в переменных **a** и **b** записаны возрасты Андрея и Бориса. Кто из них старше?

```
if a > b:
    print("Андрей старше")
else:
    if a == b:
        print("Одного возраста")
    else:
        print("Борис старше")
```



Зачем нужен?

вложенный  
условный оператор

# Каскадное ветвление

---

```
if a > b:  
    print("Андрей старше")  
elif a == b:  
    print("Одного возраста")  
else:  
    print("Борис старше")
```



**elif = else if**

# Каскадное ветвление

```
cost = 1500
if cost < 1000:
    print ( "Скидок нет." )
elif cost < 2000:
    print ( "Скидка 2%." )
elif cost < 5000:
    print ( "Скидка 5%." )
else:
    print ( "Скидка 10%." )
```

первое сработавшее  
условие



Что выведет?

Скидка 2%.

# Сложные условия

Задача: набор сотрудников в возрасте **25-40 лет**  
(включительно).

сложное условие

```
if v >= 25 and v <= 40 :  
    print ("подходит")  
else:  
    print ("не подходит")
```

**and** «И»

**or** «ИЛИ»

**not** «НЕ»

## Приоритет :

- 1) отношения (<, >, <=, >=, ==, !=)
- 2) **not** («НЕ»)
- 3) **and** («И»)
- 4) **or** («ИЛИ»)