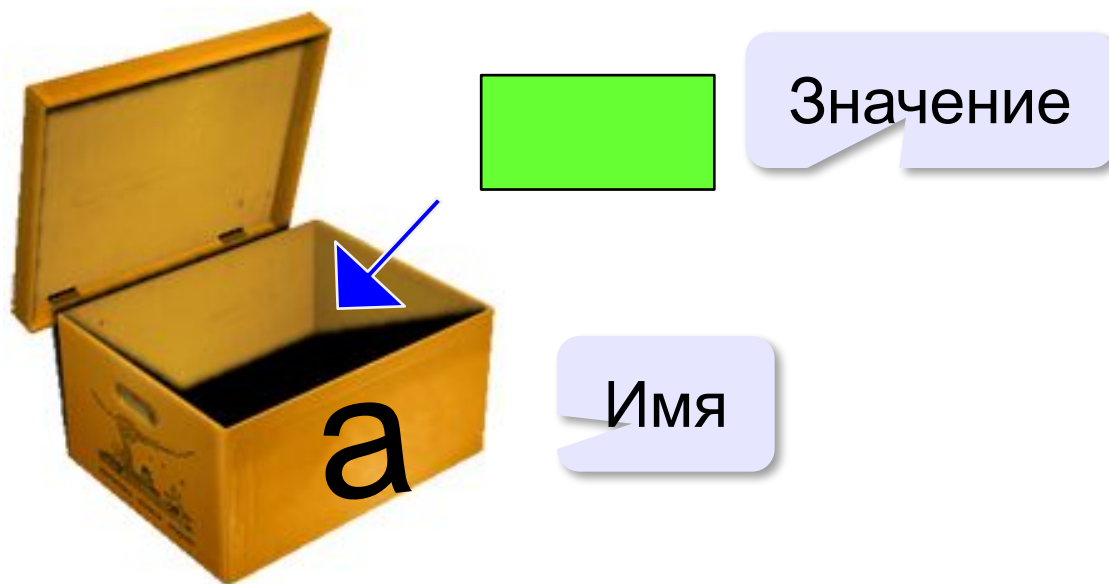


Программирование на языке Python

Простейшие программы

Переменные

Переменная — это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имена переменных

МОЖНО использовать

- латинские буквы (A-Z, a-z)

заглавные и строчные буквы **различаются**

- русские буквы (**не рекомендуется!**)
- цифры

имя не может начинаться с цифры

- знак подчеркивания _

НЕЛЬЗЯ использовать

~~• скобки~~

~~• знаки +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася “PesBarbos”
TU154 [QuQu] _ABBA A+B

Зачем нужен тип переменной?

Тип определяет:

- область допустимых значений
- допустимые операции
- объём памяти
- формат хранения данных

Типы переменных

```
a = 4  
print ( type(a) )  
<class 'int'>
```

целое число (*integer*)

```
a = 4.5  
print ( type(a) )  
<class 'float'>
```

вещественное число

```
a = "Вася"  
print ( type(a) )  
<class 'str'>
```

символьная строка

```
a = True  
print ( type(a) )  
<class 'bool'>
```

логическая

Как записать значение в переменную?

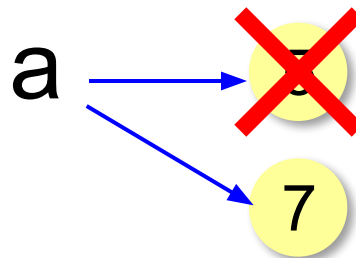
оператор
присваивания



При записи нового значения
старое удаляется из памяти!

a = 5

a = 7



Оператор – это команда языка программирования (инструкция).

Оператор присваивания – это команда для присваивания нового значения переменной.

Ввод значения с клавиатуры

```
a = input ()
```

ввести строку с клавиатуры
и связать с переменной a

```
b = input ()
```

```
c = a + b
```

```
print ( c )
```

Протокол:

21

33

2133



Почему?



Результат функции `input` – строка символов!

преобразовать в
целое число

```
a = int ( input () )
```

```
b = int ( input () )
```

Ввод с подсказкой

```
a = input ( "Введите число: " )
```

Введите число: 26

подсказка



Что не так?

```
a = int( input("Введите число: ") )
```


Вывод данных

```
print ( a )
```

значение
переменной

```
print ( "Ответ: ", a )
```

значение и
текст

перечисление через запятую

```
print ( "Ответ: ", a+b )
```

вычисление
выражения

```
print ( a, "+", b, "=", c )
```

2 + 3 = 5

через пробелы

```
print ( a, "+", b, "=", c, sep = " " )
```

2+3=5

убрать разделители

Вывод данных через `format`

```
print ( a, "+", b, "=", c, sep = " " )
```

2+3=5

```
print ( " {}+{}={} ".format(a, b, c) )
```

Сложение чисел: полное решение

```
print ( "Введите два числа: " )  
a = int ( input() )  
b = int ( input() )  
c = a + b  
print ( a, "+", b, "=", c, sep=" " )
```

подсказка

Протокол:

компьютер

Введите два целых числа

25

30

пользователь

25+30=55

Арифметическое выражения

a = (c + b5*3 - 1) / 2 * d**

Приоритет (старшинство):

1) скобки

2) возведение в степень **

3) умножение и деление

4) сложение и вычитание

$$a = (c + b * 5 * 3 - 1) \setminus / 2 * d$$
$$a = (c + b * 5 * 3 - 1) / 2 * d$$

$$a = \frac{c + b^5 \cdot 3 - 1}{2} \cdot d$$

перенос на
следующую строку

перенос внутри скобок разрешён

Операторы // (целая часть от деления) и % (остаток от деления)

```
a = 1234
```

```
d = a % 10; print( d )
```

4

```
a = a // 10
```

```
d = a % 10; print( d )
```

3

```
a = a // 10
```

```
d = a % 10; print( d )
```

2

```
a = a // 10
```

```
d = a % 10; print( d )
```

1

```
a = a // 10 :
```

Сокращенная запись операций

$a += b$	#	$a = a + b$
$a -= b$	#	$a = a - b$
$a *= b$	#	$a = a * b$
$a /= b$	#	$a = a / b$
$a // = b$	#	$a = a // b$
$a \% = b$	#	$a = a \% b$

$a += 1$

увеличение на 1

Ввод двух значений в одной строке

```
a, b = map ( int, input().split() )
```

21 33

`input()`

ввести строку с клавиатуры

21 33

`input().split()`

целые

применить

разделить строку на части по пробелам

21 33

`map (int, input().split())`

эту операцию

к каждой части

```
a, b = map ( int, input().split() )
```

Генератор случайных чисел

```
import random
```

англ. *random* – случайный

Целые числа на отрезке $[a, b]$:

```
X = random.randint(1, 6) # псевдосл. число  
Y = random.randint(1, 6) # уже другое число!
```

Генератор на $[0, 1)$:

```
X = random.random() # псевдосл. число  
Y = random.random() # уже другое число!
```

Генератор на $[a, b]$ (вещественные числа):

```
X = random.uniform(1.2, 3.5)  
Y = random.uniform(1.2, 3.5)
```


Генератор случайных чисел

```
from random import *
```

подключить все!

англ. *random* – случайный

Целые числа на отрезке [a,b]:

```
X = randint(10, 60) # псевдослучайное число  
Y = randint(10, 60) # это уже другое число!
```

Генератор на [0,1):

```
X = random() # псевдослучайное число  
Y = random() # это уже другое число!
```

Обмен случайными трехзначными числами: полное решение

```
import random
a = random.randint(100, 999)
b = random.randint(100, 999)
print("a=", a, "b=", b)
c = a
a = b
b = c
print("a=", a, "b=", b)
```

Обмен случайными трехзначными числами: полное решение

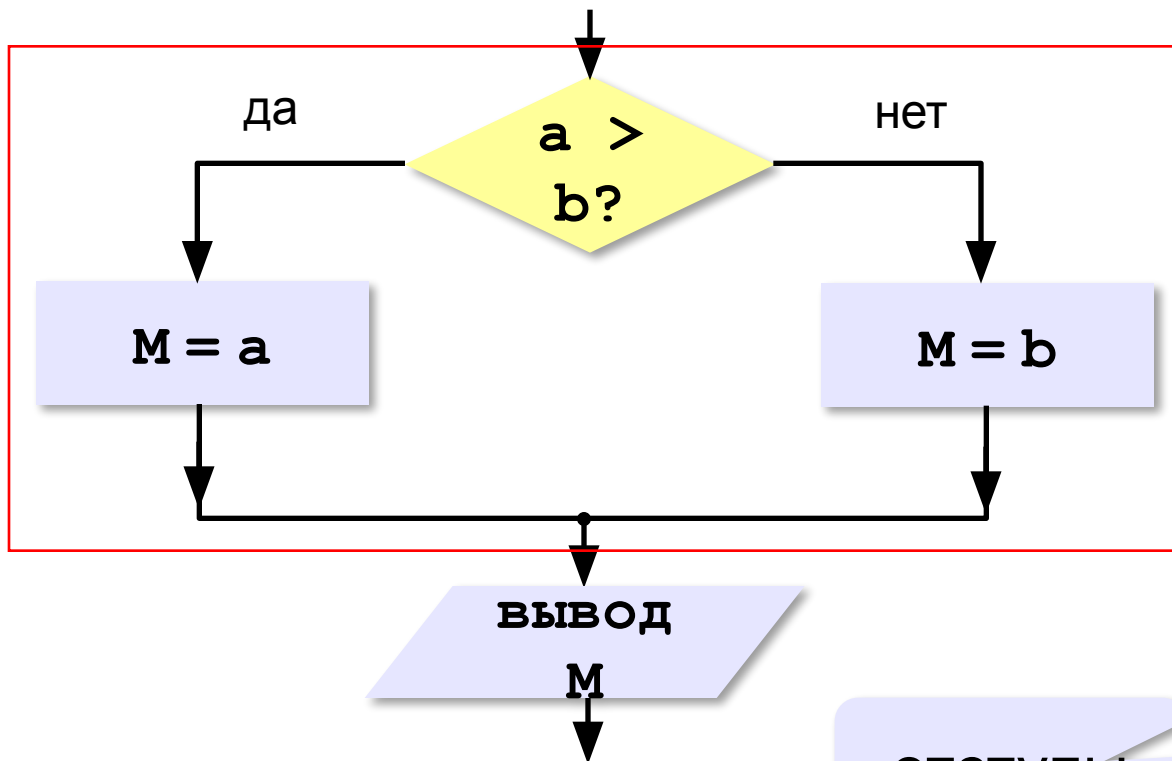
```
import random
a = random.randint(100, 999)
b = random.randint(100, 999)
print("a=", a, "b=", b)
a = a + b
b = a - b
a = a - b
print("a=", a, "b=", b)
```

Решение в стиле Python:

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

Условный оператор

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



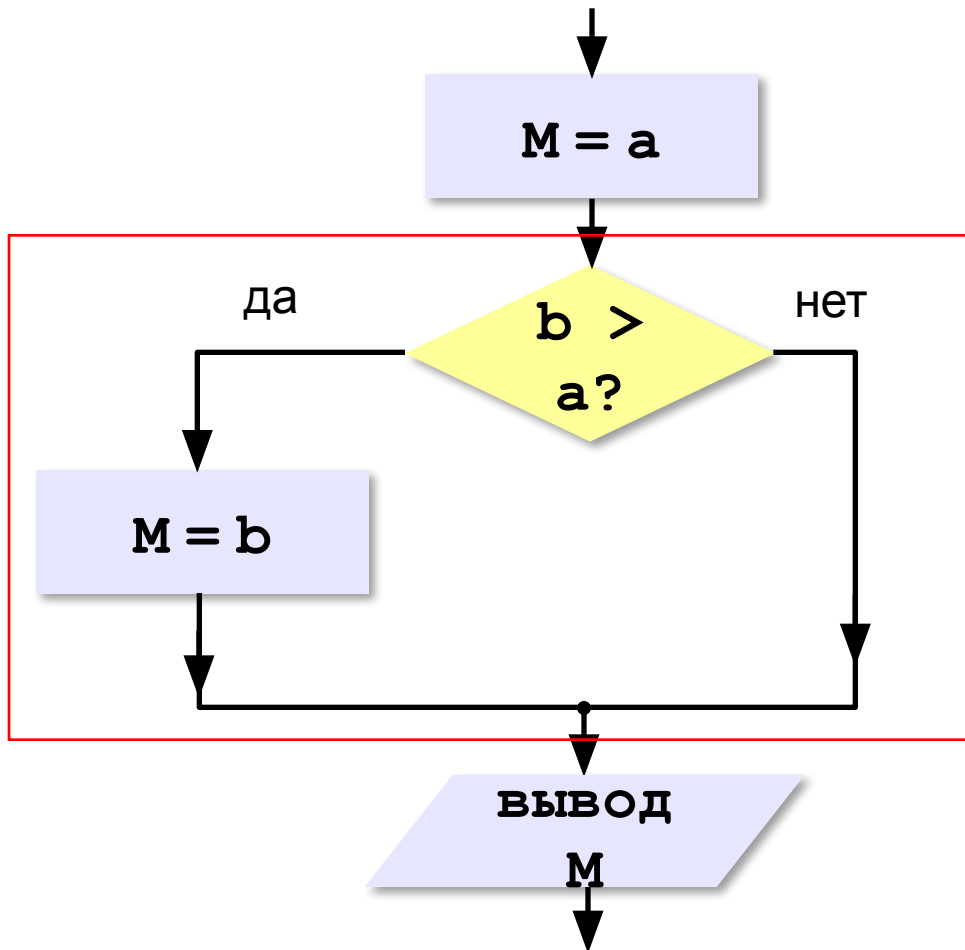
полная
форма
ветвления

? 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
```



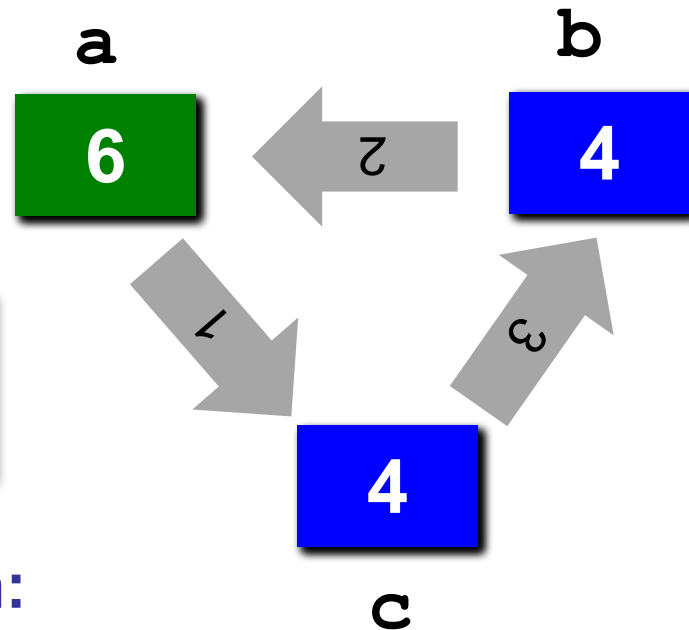
Что делает?



Можно ли обойтись без переменной **c**?

Решение в стиле Python:

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



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?

?

Сколько вариантов?

```
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»: одновременное выполнение
всех условий!

Сложные условия

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

сложное условие

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

or «ИЛИ»: выполнение **хотя бы одного**
из двух условий!

Сложные условия

```
if not (a < b):  
    print("Старт!")
```



Как без «НЕ»?

not «НЕ»: если выполняется обратное условие

```
if a >= b:  
    print("Старт!")
```

Приоритет :

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