

Программирование на языке Python

Введение в язык Python

Простейшая программа

```
# Это пустая программа
```

? Что делает эта программа?

комментарии после #
не обрабатываются

кодировка utf-8
(по умолчанию)

```
# -*- coding: utf-8 -*-
```

```
# Это пустая программа
```

Windows: cp1251

```
"""
```

```
Это тоже комментарий
```

```
"""
```

Вывод на экран

```
▶ print ( "2+2=?" )  
▶ print ( "Ответ: 4" )
```

автоматический
переход на новую
строку

Протокол:

2+2=?

Ответ: 4

```
print ( '2+2=?' )  
print ( 'Ответ: 4' )
```

Сложение чисел

Задача. Ввести с клавиатуры два числа и найти их сумму.

Протокол:

Введите два целых числа

25 30

25+30=55

компьютер

пользователь

компьютер считает сам!

?

1. Как ввести числа в память?
2. Где хранить введенные числа?
3. Как вычислить?
4. Как вывести результат?

Сумма: псевдокод

ввести два числа

вычислить их сумму

вывести сумму на экран

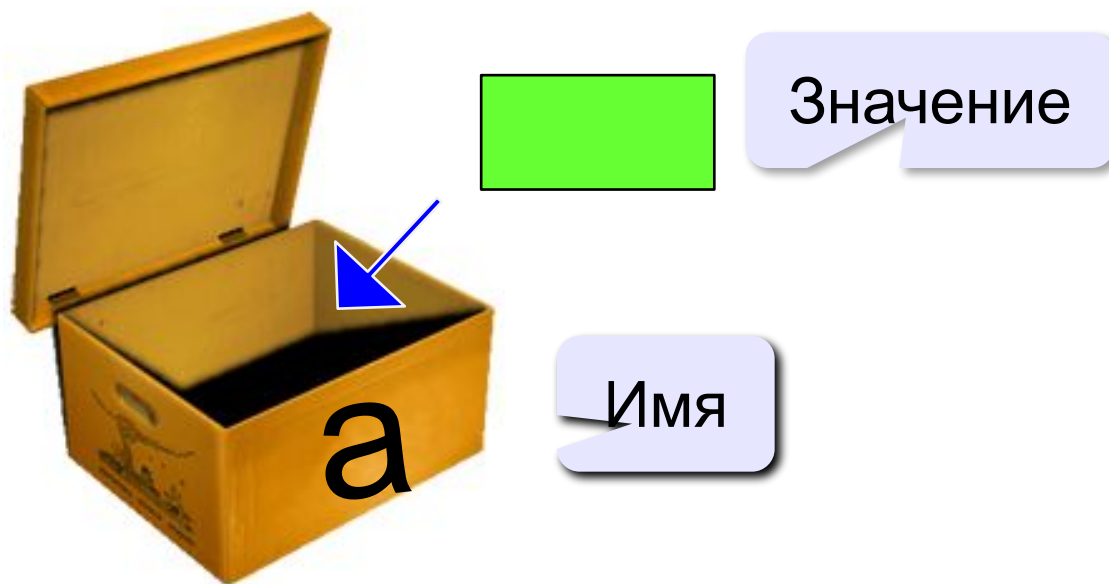
Псевдокод – алгоритм на русском языке с элементами языка программирования.



Компьютер не может исполнить псевдокод!

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

Как записать значение в переменную?

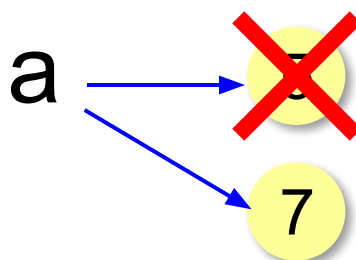
оператор
присваивания



При записи нового значения
старое удаляется из памяти!

```
a = 5
```

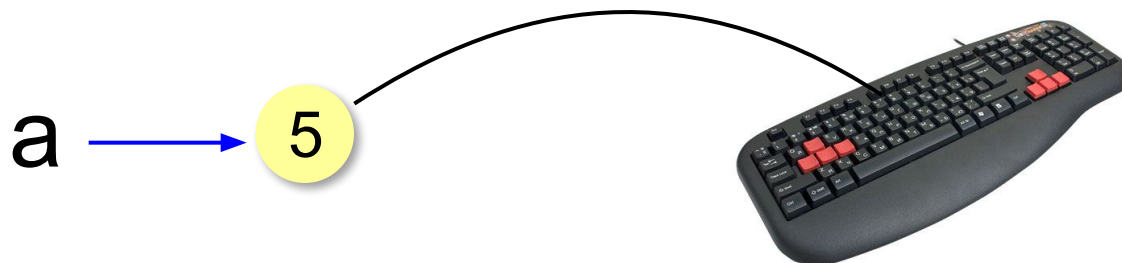
```
a = 7
```



Оператор – это команда языка программирования (инструкция).

Оператор присваивания – это команда для записи нового значения переменной.

Ввод значения с клавиатуры



1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a** (связывается с именем **a**)

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 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() )
```

Ввод с подсказкой

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

Введите число: 26

подсказка

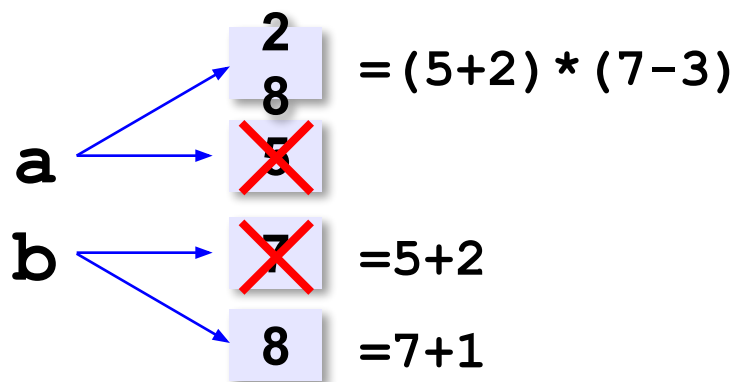


Что не так?

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

Изменение значений переменной

```
a = 5
b = a + 2
a = (a + 2) * (b - 3)
b = b + 1
```



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

Сложение чисел: простое решение

```
a = int ( input ( ) )  
b = int ( input ( ) )  
c = a + b  
print ( c )
```



Что плохо?

Сложение чисел: полное решение

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

подсказка

Протокол:

КОМПЬЮТЕР

Введите два целых числа

25 30

ПОЛЬЗОВАТЕЛЬ

25 + 30 = 55

ФОРМАТНЫЙ ВЫВОД

целое

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

```
a = 5
```

```
print ("{:5d}{:5d}{:5d}".format(a, a*a, a*a*a))
```

5	25	125
_ _ _ _	_ _ _ _	_ _
└──────────┘ └──────────┘ └──────────┘		
5 знаков 5 знаков 5 знаков		

Типы переменных

- `int` # целое
- `float` # вещественное
- `bool` # логические значения
- `str` # символьная строка

```
a = 5
```

```
print ( type (a) )
```

```
a = 4.5
```

```
print ( type (a) )
```

```
a = True
```

```
print ( type (a) )
```

```
a = "Вася"
```

```
print ( type (a) )
```

```
<class 'int'>
```

```
<class 'float'>
```

```
<class 'bool'>
```

```
<class 'str'>
```

Зачем нужен тип переменной?

Тип определяет:

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

Размещение переменных в памяти

оператор
присваивания



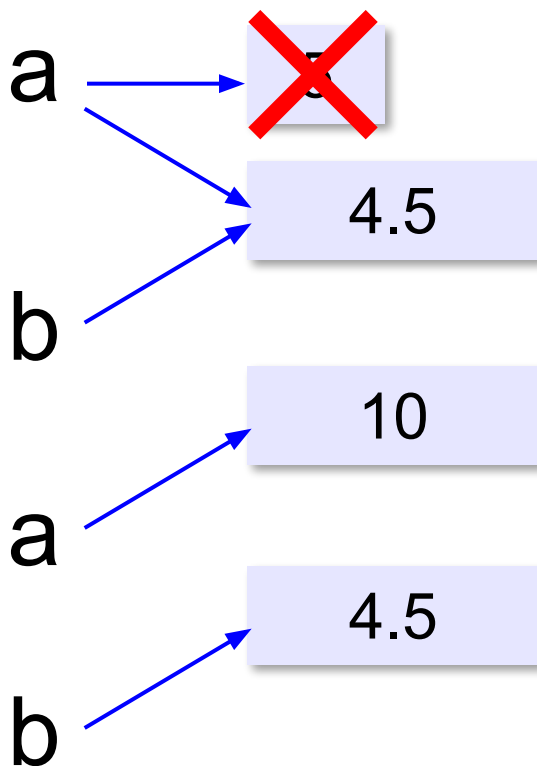
При записи нового значения
старое удаляется из памяти!

```
a = 5
```

```
a = 4.5
```

```
b = a
```

```
a = 10
```



«сборщик
мусора»

Арифметические выражения

3 1 2 4 5 6

```
a = (c + b**5*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) \
      / 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

Деление

Классическое деление:

```
a = 9; b = 6
x = 3 / 4      # = 0.75
x = a / b      # = 1.5
x = -3 / 4     # = -0.75
x = -a / b     # = -1.5
```

Целочисленное деление (округление «вниз»!):

```
a = 9; b = 6
x = 3 // 4     # = 0
x = a // b     # = 1
x = -3 // 4    # = -1
x = -a // b    # = -2
```

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

Множественное присваивание:

```
a = b = 0 # b = 0, a = b
```

```
a, b = 1, 2 # a = 1; b = 2
```

Программирование на языке Python

Вычисления

Остаток от деления

`%` – остаток от деления

```
d = 85
b = d // 10
a = d % 10
d = a % b
d = b % a
```

Для отрицательных чисел:

```
a = -7
b = a // 2 # -4
d = a % 2 # 1
```



Как в математике!

остаток ≥ 0

$$-7 = (-4) * 2 + 1$$

Вещественные числа



Целая и дробная части числа разделяются точкой!

Форматы вывода:

```
x = 123.456
```

```
print( x )
```

```
123.456
```

```
print( "{:10.2f}".format( x ) )
```

```
_____ 123.46
```

всего знаков

в дробной части

```
print( "{:10.2g}".format( x ) )
```

```
_____ 1.2e+02
```

значащих цифр

1,2 · 10²

Вещественные числа

Экспоненциальный формат:

```
x = 1. / 30000
```

```
print("{:e}".format(x))
```

```
x = 12345678.
```

```
print("{:e}".format(x))
```

$3,333333 \cdot 10^{-5}$

3.333333e-05

1.234568e+07

```
x = 123.456
```

```
print("{:e}".format(x))
```

```
print("{:10.2e}".format(x))
```

$1,234568 \cdot 10^7$

1.234560e+02

__1.23e+02

всего знаков

в дробной части

Стандартные функции

abs (x) — модуль числа

int (x) — преобразование к целому числу

round (x) — округление

```
x = abs ( -1.6 )      # 1.6
```

```
x = int ( -1.6 )     # -1
```

```
x = round ( -1.6 )   # -2
```

bin (x) — в двоичную систему

oct (x) — в восьмеричную систему

hex (x) — в шестнадцатеричную систему

```
x = bin ( 29 )      # '0b11101'
```

```
x = oct ( 29 )     # '0o35'
```

```
x = hex ( 29 )     # '0x1d'
```

Математические функции

```
import math
```

ПОДКЛЮЧИТЬ
МАТЕМАТИЧЕСКИЙ МОДУЛЬ

- `math.pi` — ЧИСЛО «ПИ»
- `math.sqrt(x)` — квадратный корень
- `math.sin(x)` — синус угла, заданного **в радианах**
- `math.cos(x)` — косинус угла, заданного **в радианах**
- `math.exp(x)` — экспонента e^x
- `math.ln(x)` — натуральный логарифм
- `math.floor(x)` — округление «вниз»
- `math.ceil(x)` — округление «вверх»

```
x = math.floor(1.6) # 1  
x = math.ceil(1.6) # 2
```

```
x = math.floor(-1.6) #-2  
x = math.ceil(-1.6) #-1
```

Документирование программы

```
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?

Документирование программы

Руководство пользователя:

- назначение программы
- формат входных данных
- формат выходных данных
- примеры использования программы

Назначение:

программа 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

Документирование программы

Формат выходных данных:

значения вещественных корней уравнения;
если вещественных корней нет, выводится
слово «нет»

Примеры использования программы:

1. Решение уравнения $x^2 - 5x + 1 = 0$

Введите a, b, c: **1 -5 1**

x1=4.791 x2=0.209

2. Решение уравнения $x^2 + x + 6 = 0$

Введите a, b, c: **1 1 6**

Нет.

Случайные числа

Случайно...

- встретить друга на улице
- разбить тарелку
- найти 10 рублей
- выиграть в лотерею

Случайный выбор:

- жеребьевка на соревнованиях
- выигравшие номера в лотерее

Как получить случайность?



Случайные числа на компьютере

Электронный генератор



- нужно специальное устройство
- нельзя воспроизвести результаты

Псевдослучайные числа – обладают свойствами случайных чисел, но каждое следующее число вычисляется по заданной формуле.

Метод середины квадрата (Дж. фон Нейман)

зерно

564321

в квадрате

- малый период
(последовательность повторяется через 10^6 чисел)

318458191041

209938992481

Линейный конгруэнтный генератор

$$X = (a * X + b) \% c \quad | \quad \text{интервал от } 0 \text{ до } c-1$$

$$X = (X + 3) \% 10 \quad | \quad \text{интервал от } 0 \text{ до } 9$$

$$X = 0 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 2 \rightarrow 5 \rightarrow 8$$

$$8 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 0$$

зерно

заикливание



Важен правильный выбор параметров a , b и c !

Компилятор GCC:

$$a = 1103515245$$

$$b = 12345$$

$$c = 2^{31}$$

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!
```

Генератор на [0,1):

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

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!
```

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

```
X = uniform(1.2, 3.5) # псевдосл. число  
Y = uniform(1.2, 3.5) # уже другое число!
```

Задачи

«А»: Ввести с клавиатуры три целых числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три целых числа:

5 7 8

$$5+7+8=20$$

$$5*7*8=280$$

$$(5+7+8)/3=6.667$$

«В»: Ввести с клавиатуры координаты двух точек (А и В) на плоскости (вещественные числа). Вычислить длину отрезка АВ.

Пример:

Введите координаты точки А:

5.5 3.5

Введите координаты точки В:

1.5 2

$$\text{Длина отрезка АВ} = 4.272$$

Задачи

«С»: Получить случайное трехзначное число и вывести через запятую его отдельные цифры.

Пример:

Получено число 123.

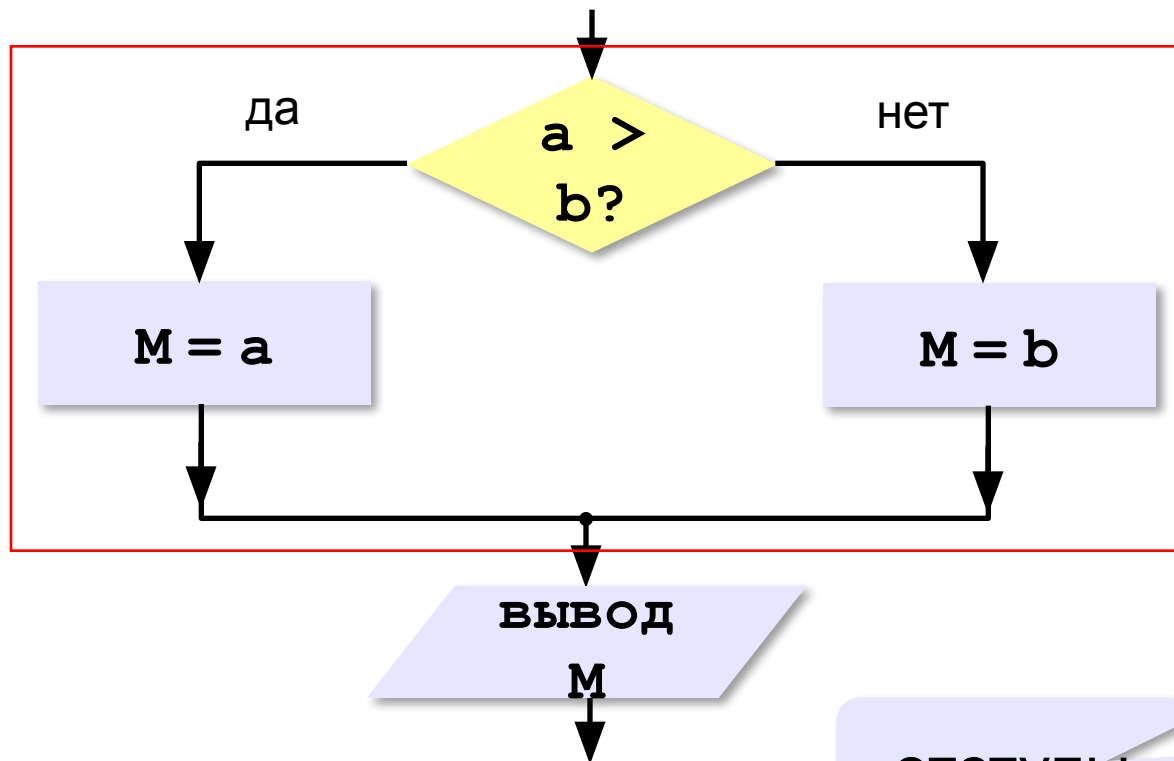
Его цифры 1, 2, 3.

Программирование на языке Python

Ветвления

Условный оператор

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



полная
форма
ветвления

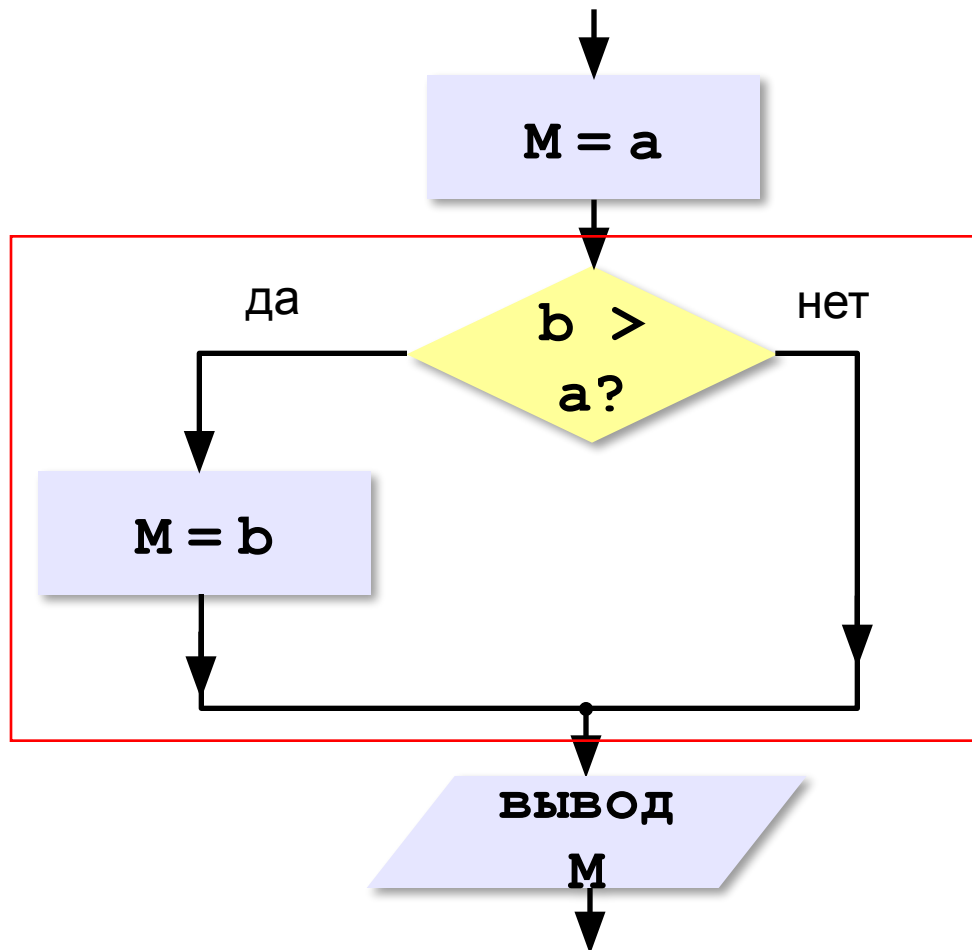


Если $a = b$?

```
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

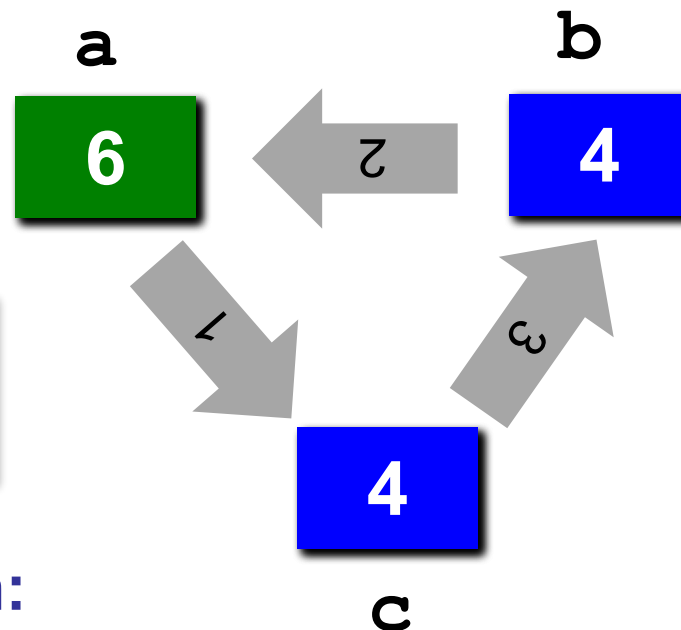
```
if a < b:
```

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



Зачем нужен?

вложенный
условный оператор

Каскадное ветвление

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

Задачи

«А»: Ввести три целых числа, найти максимальное из них.

Пример:

Введите три целых числа:

1 5 4

Максимальное число 5

«В»: Ввести пять целых чисел, найти максимальное из них.

Пример:

Введите пять целых чисел:

1 5 4 3 2

Максимальное число 5

Задачи

«С»: Ввести последовательно возраст Антона, Бориса и Виктора. Определить, кто из них старше.

Пример:

Возраст Антона: 15

Возраст Бориса: 17

Возраст Виктора: 16

Ответ: Борис старше всех.

Пример:

Возраст Антона: 17

Возраст Бориса: 17

Возраст Виктора: 16

Ответ: Антон и Борис старше Виктора.

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»)

Задачи

«А»: Напишите программу, которая получает три числа и выводит количество одинаковых чисел в этой цепочке.

Пример:

Введите три числа:

5 5 5

Все числа одинаковые.

Пример:

Введите три числа:

5 7 5

Два числа одинаковые.

Пример:

Введите три числа:

5 7 8

Нет одинаковых чисел.

Задачи

«В»: Напишите программу, которая получает номер месяца и выводит соответствующее ему время года или сообщение об ошибке.

Пример:

Введите номер месяца :

5

Весна .

Пример:

Введите номер месяца :

15

Неверный номер месяца .

Задачи

«С»: Напишите программу, которая получает возраст человека (целое число, не превышающее 120) и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «25 лет».

Пример:

Введите возраст: **18**

Вам 18 лет.

Пример:

Введите возраст: **21**

Вам 21 год.

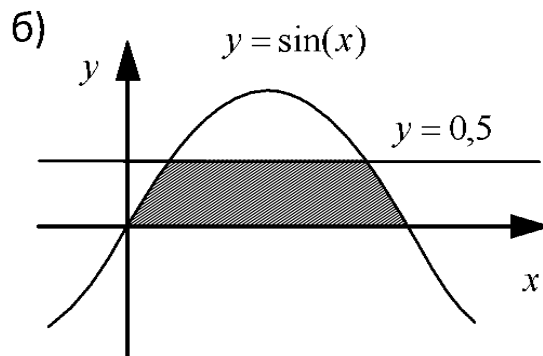
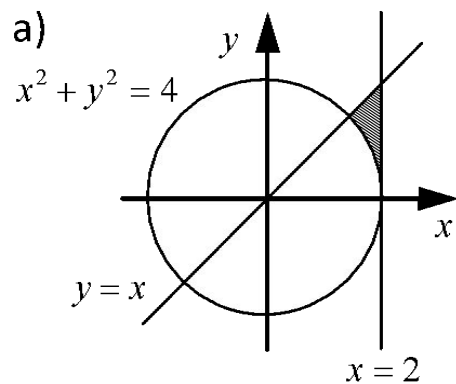
Пример:

Введите возраст: **22**

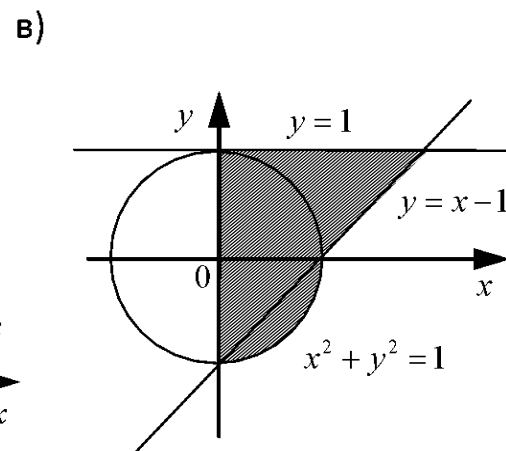
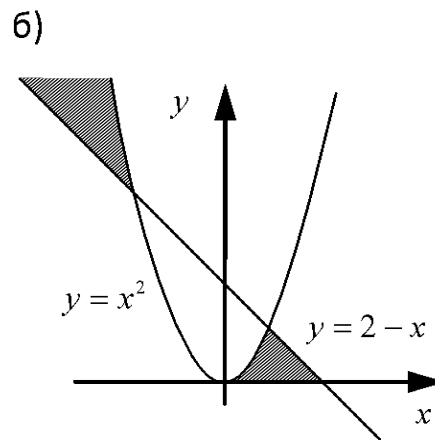
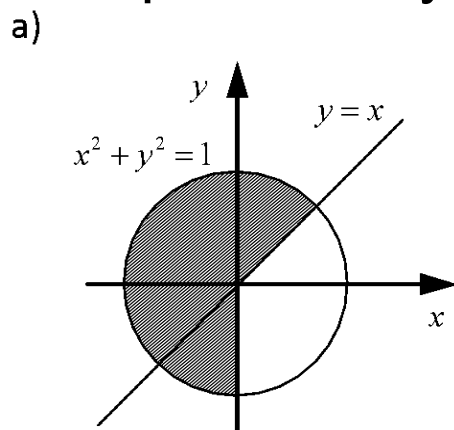
Вам 22 года.

Задачи

«А»: Напишите условие, которое определяет заштрихованную область.



«В»: Напишите условие, которое определяет заштрихованную область.



Задачи

«С»: Напишите условие, которое определяет заштрихованную область.

