

# Язык программирования Python

---



# Язык Python



Для создания файла с программой в меню File нужно выбрать New File. В открывшемся окне набрать текст программы (скрипт), сохранить его под каким-нибудь именем в меню File → Save As, запустить на выполнение в меню Run → Run Module или нажав клавишу F5.

Результат работы программы отобразится в окне Python Shell.

```
File Edit Format Run Options Window Help
print("О, сколько нам открытий чудных")
print("Готовят просвещения дух")
print("И опыт, сын ошибок трудных,")
print("И гений, парадоксов друг...")
Ln: 4 Col: 31
```

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\program.py =====
О, сколько нам открытий чудных
Готовят просвещения дух
И опыт, сын ошибок трудных,
И гений, парадоксов друг...
>>> |
Ln: 9 Col: 4
```

Пример работы в программном режиме



# Общие сведения о языке программирования Python

**Алфавит** языка Python (набор допустимых символов) состоит из

- букв латинского алфавита (причём *заглавные и строчные буквы различаются*),
- цифр и специальных знаков (знаков препинания, арифметических и других).

Русские буквы могут использоваться только при выводе текста на экран и в комментариях к программе.

**Служебные слова** – цепочки символов, имеющие фиксированное смысловое значение.

**Величины** в программе представлены в виде констант и переменных.

**Константы** – величины, не изменяющие своего значения при выполнении программы.

**Переменные** – величины, которые могут изменять свое значение при выполнении программы. Каждая переменная имеет имя, тип и значение.

**Имя переменной (идентификатор)** – любая отличная от служебных слов последовательность латинских букв, цифр и символа подчеркивания "\_", **не может начинаться с цифры.**

N, N1, massa, massa\_tela – **правильно**;

1N, масса, massa tela – **неправильно.**



# Выражения и операции

**Выражение** – это конструкция, возвращающая значение некоторого типа.

Простыми выражениями являются переменные и константы.

Сложные выражения строятся из простых с помощью операций, функций и скобок. Данные, к которым применяются операции, называются **операндами**.

Используется линейная форма записи выражений (в одну строку).

## Арифметические операции

Операция	Обозначение	Пример
Сложение	<b>+</b>	$3 + 4 = 7$
Вычитание	<b>-</b>	$7 - 2 = 5$
Умножение	<b>*</b>	$2 * 2 = 4$
Деление	<b>/</b>	$8 / 2 = 4$
Целочисленное деление	<b>//</b>	$9 // 2 = 4$
Остаток от деления	<b>%</b>	$9 \% 2 = 1$
Возведение в степень	<b>**</b>	$2 ** 3 = 8$



# Выражения и операции

**Логические выражения** могут содержать величины или выражения, которые сравниваются между собой с помощью операций сравнения.

Логическое выражение может принимать лишь два значения: «истина» или «ложь».

## Операции сравнения

Операция	Символы	Пример
равно	<code>==</code>	<code>x == 0</code>
не равно	<code>!=</code>	<code>x != 0</code>
больше	<code>&gt;</code>	<code>x &gt; 0</code>
меньше	<code>&lt;</code>	<code>x &lt; 0</code>
больше или равно	<code>&gt;=</code>	<code>x &gt;= 0</code>
меньше или равно	<code>&lt;=</code>	<code>x &lt;= 0</code>



# Выражения и операции

## Приоритет выполнения операций:

- 1) операции в скобках;
- 2) возведение в степень;
- 3) умножение и деление (в том числе // и %);
- 4) сложение и вычитание.

Операции одинакового приоритета выполняются в порядке записи слева направо.

Если выражение слишком длинное и не помещается в одной строке, необходимо заключить всё выражение в скобки (перенос внутри скобок разрешён).

## Например:

$$\frac{(a+b)h}{2} \quad \longrightarrow \quad (a+b) * h / 2$$

The diagram shows the conversion of a mathematical fraction to a Python expression. A blue arrow points from the fraction  $\frac{(a+b)h}{2}$  to the Python expression  $(a+b) * h / 2$ . Above the Python expression, red numbers 1, 2, and 3 are placed above the parentheses, the multiplication sign, and the division sign, respectively, to indicate the order of operations.

$$v + \frac{at^2}{2} \quad \longrightarrow \quad v + a * t ** 2 / 2$$

The diagram shows the conversion of a mathematical expression with a fraction to a Python expression. A blue arrow points from the expression  $v + \frac{at^2}{2}$  to the Python expression  $v + a * t ** 2 / 2$ . Above the Python expression, red numbers 4, 2, 1, and 3 are placed above the addition sign, the multiplication sign, the exponentiation sign, and the division sign, respectively, to indicate the order of operations.

## Функция вывода

`print()` - выводит заданные объекты (числа, текст и др.) на экран

Описание функции:

```
print(*objects, sep='', end='')
```

- `objects` - объект, который нужно вывести (\* обозначает, что объектов может быть несколько);
- `sep` - указываем разделитель между объектами (значение по умолчанию - пробел);
- `end` - указываем символ, который ставится после всех объектов.

## Функция ввода

`a = input()` - читает данные с клавиатуры и записывает в переменную `a`

`a = int(input())` - чтение целого числа с клавиатуры

`a = float(input())` - чтение дробного числа с клавиатуры

# Типы данных

## Числовые

- **int** - целые числа
- **float** - вещественные числа

## Строковые

- **str** - строки



# Оператор ввода

Для ввода значений переменных с клавиатуры в процессе выполнения программы используется оператор (функция) ввода **input** («ввод»):

```
<имя_переменной> = input()
```

При выполнении оператора:

- компьютер переходит в режим ожидания данных;
- пользователь вводит с клавиатуры данные в виде строки символов;
- для завершения ввода пользователь нажимает клавишу Enter;
- введенная строка записывается в указанную переменную.

Если вводится не строка, а число, необходимо выполнить преобразование типов с помощью функций **int** (для целых) и **float** (для вещественных).

*Например:*

*На экране:*

```
print("Введите слово и два числа:")  
x = input()  
y = int(input())  
z = float(input())  
print(x, y, z)
```

```
Введите слово и два числа:  
ноль  
1  
2  
ноль 1 2.0
```



## Вычислить

$$45678 \% 10 = 8$$

$$45678 // 1000 = 45$$

$$45678 \% 100 = 78$$

$$45678 // 10000 = 4$$

$$1234 \% 100 // 10 = 3$$



# Задача 1

Составить программу, вычисляющую сумму цифр введенного с клавиатуры целого трёхзначного числа

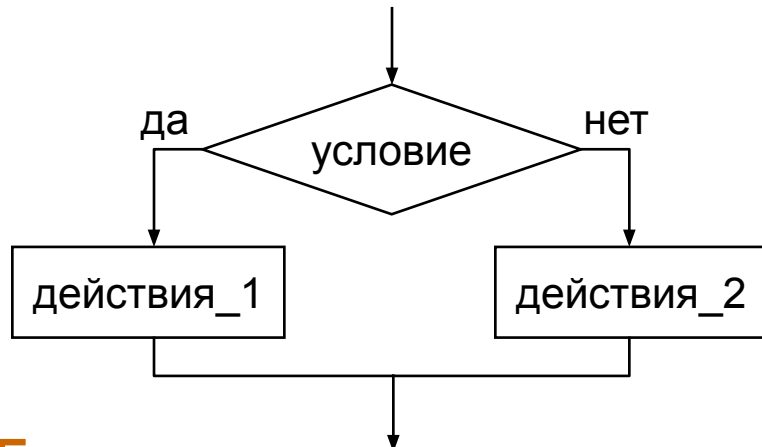
```
# Сумма цифр трехзначного числа
# Трёхзначное число  $x = a \cdot 100 + b \cdot 10 + c$ 
# где  $a, b, c$  - цифры этого числа
x = int(input("Введите трехзначное число: "))
a = x // 100          # сотни
b = x % 100 // 10    # десятки
c = x % 10           # единицы
s = a + b + c
print ("Сумма цифр равна", s)
```

На экране:

```
Введите трехзначное число: 345
Сумма цифр равна 12
```



## Полная форма ветвления (условного оператора):



**Если** условие истинно, то выполняется блок\_1, **иначе** – блок\_2.

В языке Python важную роль играют отступы операторов от левой границы текста программы. Начало и конец блоков операторов определяется этими сдвигами. Задаются отступы пробелами или клавишей **Tab**.

Если в блоке всего один оператор, то допустима такая форма записи:

```
if <условие>: <оператор_1>
else: <оператор_2>
```

```
if условие1:
    действие 1
elif условие2:
    действие 2
elif условие3:
    действие 3
...
else:
    действие 4
```



## Часто используемые условия

Условие	Что проверяет
<code>if a &gt; 0:</code>	число $a$ положительное
<code>if a &gt;= 0:</code>	число $a$ неотрицательное
<code>if a &lt; 0:</code>	число $a$ отрицательное
<code>if a &lt;= 0:</code>	число $a$ неположительное
<code>if a % 2 == 0:</code>	число $a$ чётное
<code>if a % 2 != 0:</code>	число $a$ нечётное
<code>if a % 3 == 0:</code>	число $a$ кратно 3
<code>if a % 10 == 6:</code>	число $a$ оканчивается на 6
<code>if a % 100 == 44:</code>	число $a$ оканчивается на 44

# Пример программы



Напишите программу, которая определяет, равны ли два числа между собой, и выводит соответствующий комментарий.

## **Входные данные**

Вводится два целых числа, по модулю не превышающих 10 000 (то есть от -10 000 до 10 000).

## **Выходные данные**

Вывести комментарий "равны", если числа равны между собой. В противном случае вывести комментарий "не равны".

### **Sample Input 1:**

6 8

### **Sample Output 1:**

не равны

### **Sample Input 2:**

7 7

### **Sample Output 2:**

равны

## *Решение вариант 1*

```
a=float(input())
b=float(input())
if a==b:
    print("равны")
else:
    print("не равны")
```

## *Решение вариант 2*

```
a=int(input())
b=int(input())
if a==b:
    print('равны')
else:
    print('не равны')
```

# Составное условие



## Операция **and**

выполняются

**ОБА**

условия одновременно

В данном примере условие будет верно, если  $a > 0$  и  $b < 0$

```
if (a>0) and (b<0):  
    print(a,b)
```

## Операция **or**

выполняется

**ХОТЯ БЫ ОДНО**

из условий

А здесь условие будет верно, если  $a > 0$  или  $b < 0$ , т.е. верно хотя бы одно из условий

```
if (a>0) or (b<0):  
    print(a,b)
```

Также можно составлять различные комбинации из условий с использованием скобок.  
Например:

```
if ((a>0) or (b<0)) and (c==0):  
    print(a,b)
```



# Задача 1

**Напишите программу, которая определяет, сколько разрядов в числе и выводит один из трёх комментариев:**

«число однозначное»

«число двузначное»

«число трехзначное»

## **Входные данные**

Вводится одно натуральное число меньше 1000.

## **Выходные данные**

Программа должна вывести один из трёх комментариев.

### **Sample Input:**

23

### **Sample Output:**

число двузначное

```
t = int(input())
if t <= 9:
    print('число однозначное')
if t <= 99 and t >= 10:
    print('число двузначное')
if t >= 100 and t < 1000:
    print('число трехзначное')
```