
Программирование на языке Python

Логические выражения и операторы сравнения.

Логический оператор

**Программирование выражений и функций в
линейных алгоритмах на языке Python**

Программирование алгоритмов ветвления на языке

**Python
Программирование циклических алгоритмов на языке
программирования Python**

Операторы в Python

- ❑ Операторами пользуются для выполнения операций с переменными и значениями.
- ❑ **Python делит операторы на следующие группы:**
 - ✓ Арифметические операторы
 - ✓ Операторы присваивания
 - ✓ Операторы сравнения
 - ✓ Логические операторы

Арифметические операторы в Python

Арифметические операторы используются с числовыми значениями для выполнения общих математических операций:

Оператор	Значение
+	добавление
-	вычитание
*	умножение
/	деление
%	остаток от деления
**	возведение в степень
//	целочисленное деление

Пример
4 + 5
8 - 3
5 * 5
4 / 2
7 % 2
2 ** 3
15 // 4

Операторы присваивания в Python

Операторы присваивания
используются для
присваивания значений
переменным:



Оператор
=
+=
-=
*=
/=
%=
//=
**=
&=
=

Пример	Так же как
x = 5	x = 5
x += 3	x = x + 3
x -= 3	x = x - 3
x *= 3	x = x * 3
x /= 3	x = x / 3
x %= 3	x = x % 3
x //= 3	x = x // 3
x **= 3	x = x ** 3
x &= 3	x = x & 3
x = 3	x = x 3

Операторы сравнения в Python

Операторы сравнения используются для сравнения двух значений:

Оператор	Значение
<code>==</code>	равно
<code>!=</code>	не равно
<code>></code>	больше чем
<code><</code>	меньше чем
<code>>=</code>	больше чем или равно
<code><=</code>	меньше чем или равно

Пример
<code>x == y</code>
<code>x != y</code>
<code>x > y</code>
<code>x < y</code>
<code>x >= y</code>
<code>x <= y</code>

Логические операторы в Python

Логические операторы используются для объединения условных операторов:

Оператор	Значение
and	Возвращает значение True если оба утверждения верны
or	Возвращает True если одно из утверждений верно
not	Меняет результат, возвращает False если результат True

Пример
<code>x < 5 and x < 10</code>
<code>x < 5 or x < 4</code>
<code>not(x < 5 and x < 10)</code>

Напишите программу, которая выводит на экран текст «I***like***Python» (без кавычек).

```
print ("I", "like", "Python", sep="***")
```

Результат:

I***like***Python

sep — это может быть строка, которую необходимо вставлять между значениями, по умолчанию — пробел.

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b4
AMD64)] on win32
Type "help", "copyright", "cre
>>>
===== RESTART: C:/Users/U
1
2
3
6
>>>
===== RESTART: C:/Users/U
I***like***Python
>>>
===== RESTART: C:/Users/U
I like Python
>>>
===== RESTART: C:/Users/U
I, like, Python
>>>
===== RESTART: C:/Users/U
I, like, Python
>>>
===== RESTART: C:/Users/U
I like Python
>>> |
```

Сумма трёх чисел

Напишите программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке.

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

Результат:

1

2

3

6

Следующее и предыдущее

Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом.

```
a = int(input())  
print('Следующее за числом', a, 'число:', a+1)  
print('Для числа', a, 'предыдущее число:', a-1)
```

Результат:

10

Следующее за числом 10 число: 11
Для числа 10 предыдущее число: 9

Разделяй и властвуй

Напишите программу, которая считывает целое положительное число **xx** и выводит на экран последовательность чисел **x, 2x, 3x, 4x, 5x**, разделённых тремя черточками.

```
a = int(input())  
print(a, a*2, a*3, a*4, a*5, sep='-'*3)
```

Результат:

10

10---20---30---40---50

Расстояние в метрах

Напишите программу, которая находит полное число метров по заданному числу сантиметров.

```
a = int(input())  
print(a//100)
```

Результат:

1050

10

Сама неотвратимость

Безумный титан Танос собрал все 6 камней бесконечности и намеревается уничтожить половину населения Вселенной по щелчку пальцев. При этом если население Вселенной является нечетным числом, то титан проявит милосердие и округлит количество выживших в большую сторону. Помогите Мстителям подсчитать количество выживших.

```
a = int(input())  
print(a // 2 + a % 2)
```

Результат:

1015

508

Пересчет временного интервала

Напишите программу для пересчёта величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах.

```
a = int(input())  
print(a, 'мин - это', a // 60, 'час', a % 60, 'минут.')
```

Результат:

90

90 мин - это 1 час 30 минут

Трёхзначное число

Напишите программу, в которой рассчитывается сумма и произведение цифр положительного трёхзначного числа.

```
a = int(input())
a1 = a // 100
a2 = a % 100 // 10
a3 = a % 10
print('Сумма цифр =', a1 + a2 + a3)
print('Произведение цифр =', a1 * a2 * a3)
```

Результат:

132

Сумма цифр = 6 Произведение цифр = 6

Трёхзначное число - Альтернативное решение:

Напишите программу, в которой рассчитывается сумма и произведение цифр положительного трёхзначного числа.

```
a = str(input())  
print('Сумма цифр =', int(a[0]) + int(a[1]) + int(a[2]))  
print('Произведение цифр =', int(a[0]) * int(a[1]) * int(a[2]))
```

Результат:

132

Сумма цифр = 6 Произведение цифр = 6

Четырёхзначное число

Напишите программу для нахождения цифр четырёхзначного числа.

```
a = int(input())
print("Цифра в позиции тысяч равна", a // 1000)
print("Цифра в позиции сотен равна", a % 1000 // 100)
print("Цифра в позиции десятков равна", a % 100 // 10)
print("Цифра в позиции единиц равна", a % 10)
```

Результат:

1234

Сумма цифр = 6 Произведение цифр = 6

Пароль

При регистрации на сайтах требуется вводить пароль дважды. Это сделано для безопасности, поскольку такой подход уменьшает возможность неверного ввода пароля.

Напишите программу, которая сравнивает пароль и его подтверждение. Если они совпадают, то программа выводит: «Пароль принят», иначе: «Пароль не принят».

```
a = (input()) b = (input())  
print('Пароль принят') if a == b else  
print('Пароль не принят')
```

Результат:

123456

123456

Пароль принят

Четное или нечетное?

Напишите программу, которая определяет, является число четным или нечетным

```
a = int(input())  
print('Нечетное') if a % 2 else print('Четное')
```

Результат:

50

Четное

Арифметическая прогрессия

Напишите программу, которая определяет, являются ли три заданных числа (в указанном порядке) последовательными членами арифметической прогрессии.

```
a1 = int(input())
a2 = int(input())
a3 = int(input())
if a3 - a2 == a2 - a1:
    print('YES')
else:
    print('NO')
```

Результат:

1
2
3
YES

Наименьшее из четырёх чисел

Напишите программу, которая определяет наименьшее из четырёх чисел.

```
a, b, c, d = int(input()), int(input()), int(input()), int(input())
if a > b:
    a = b
if c > d:
    c = d
if a > c:
    a = c
print(a)
```

Результат:

1
2
3
4

1

Наименьшее из четырёх чисел / Альтернативное решение:

Напишите программу, которая определяет наименьшее из четырёх чисел.

```
a1 = int(input())  
a2 = int(input())  
a3 = int(input())  
a4 = int(input())  
print(min(a1, a2, a3, a4))
```

Результат:

1

2

3

4

1

Только +

Напишите программу, которая считывает три числа и подсчитывает сумму только положительных чисел.

```
a = list()
a.append(int(input()))
a.append(int(input()))
a.append(int(input()))
s = 0
for i in a:
    if i >= 0:
        s += i
print(s)
```

Результат:

1
2
-3

3

Что значит a.append?

Добавляет указанный элемент в конец списка. x : Элемент, который требуется добавить в список.

Только + / Альтернативное решение:

Напишите программу, которая считывает три числа и подсчитывает сумму только положительных чисел.

```
a, b, c = int(input()), int(input()), int(input())  
print((a if a > 0 else 0) + (b if b > 0 else 0) + (c if c > 0 else 0))
```

Результат:

-2

-3

5

5

Принадлежность 1

Напишите программу, которая принимает целое число **x** и определяет, принадлежит ли данное число промежутку от **-1 до 17**.

```
a1 = int(input())
if a1 > -1 and a1 < 17:
    print('Принадлежит')
else:
    print('Не принадлежит')
```

Результат:
10
Принадлежит

Принадлежность 2

Напишите программу, которая принимает целое число **x** и определяет, принадлежит ли данное число указанным промежуткам: от минус бесконечности до **-3** и **от 7** до бесконечности.

```
a1 = int(input())  
print('Принадлежит') if (a1 <= -3 or a1 >= 7) else print('Не принадлежит')
```

Результат:
10
Принадлежит

Принадлежность 3

Напишите программу, которая принимает целое число **x** и определяет, принадлежит ли данное число указанным промежуткам:
от **-30** до **-2** и от **7** до **25**.

```
x = int(input())  
print('Принадлежит') if (x > -30 and x <= -2) or (x > 7 and x <= 25) else  
print('Не принадлежит')
```

Результат:
10
Принадлежит

Високосный год

Напишите программу, которая определяет, является ли год с данным номером високосным. Если год является високосным, то выведите «YES», иначе выведите «NO». Год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400.

```
a1 = int(input())  
print('YES') if ((a1 % 4 == 0) and not (a1 % 100 == 0)) or (a1 % 400 == 0) else print('NO')
```

Результат:

2021

NO

Ход ладьи

Даны две различные клетки шахматной доски.

Напишите программу, которая определяет, может ли ладья попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от **1 до 8** каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести «**YES**», если из первой клетки ходом ладьи можно попасть во вторую, или «**NO**» в противном случае.

```
a1 = int(input())
a2 = int(input())
a3 = int(input())
a4 = int(input())
print('YES') if (a1 == a3) or (a2 == a4) else
print('NO')
```

Результат:

1

1

2

1

YES

Ход короля

Даны две различные клетки шахматной доски. Напишите программу, которая определяет, может ли король попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести «YES», если из первой клетки ходом короля можно попасть во вторую, или «NO» в противном случае.

```
a, b, c, d = (int(input()) for i in 'abcd')
if abs(a - c) <= 1 and abs(b - d) <= 1:
    print('YES')
else:
    print('NO')
```

Результат:

```
1
1
2
2
YES
```

Вид треугольника

Напишите программу, которая принимает три положительных числа и определяет вид треугольника, длины сторон которого равны введенным числам.

```
a1 = int(input())
a2 = int(input())
a3 = int(input())
if a1 == a2 == a3:
    print('Равносторонний')
elif a2 == a3 or a1 == a3 or a1 == a2:
    print('Равнобедренный')
else:
    print('Разносторонний')
```

Результат:

1

3

2

Разносторонний

Самописный калькулятор

Напишите программу, которая считывает с клавиатуры два целых числа и строку. Если эта строка является обозначением одной из четырёх математических операций (+, -, *, /), то выведите результат применения этой операции к введённым ранее числам, в противном случае выведите «Неверная операция». Если пользователь захочет поделить на ноль, выведите текст «На ноль делить нельзя!».

```
a1 = int(input())
a2 = int(input())
a3 = input()

if a3 == '/' and a2 == 0:
    print('На ноль делить нельзя!')
elif a3 == '/':
    print(a1 / a2)
elif a3 == '+':
    print(a1 + a2)
elif a3 == '-':
    print(a1 - a2)
elif a3 == '*':
    print(a1 * a2)
else:
    print('Неверная операция')
```

Результат:

3
6
+
9

Цикл for

В списке значений могут быть выражения различных типов:

```
for i in 1, 2, 3, 'one', 'two', 'three':    print(i)
```

Результат:

1

2

3

one

two

three

```
i = 1
```

```
for color in 'red', 'orange', 'yellow', 'green', 'cyan', 'blue', 'violet':  
    print('#', i, ' color of rainbow is ', color, sep = ")    i += 1
```

Цикл for, также называемый циклом с параметром, в языке Питон богат возможностями. В цикле for указывается переменная и множество значений, по которому будет пробегать переменная. Множество значений может быть задано списком, кортежем, строкой или диапазоном.

Функция range

Как правило, **циклы for** используются либо для повторения какой-либо последовательности действий заданное число раз, либо для изменения значения переменной в цикле от некоторого начального значения до некоторого конечного.

Для повторения цикла некоторое заданное число раз **n** можно использовать цикл **for** вместе с функцией **range**:

```
for i in range(4):  
    print(i)  
    print(i ** 2)  
    print('Конец цикла')
```

```
>>> 0  
0  
1  
1  
2  
4  
3  
9  
Конец цикла  
>>>  
=====
```

```
=====
```

```
0  
0  
Конец цикла  
1  
1  
1  
Конец цикла  
2  
4  
Конец цикла  
3  
9  
Конец цикла  
>>>
```

Цветовой микшер

Красный, синий и желтый называются основными цветами, потому что их нельзя получить путем смешения других цветов.

При смешивании двух основных цветов получается вторичный цвет:

- если смешать красный и синий, то получится фиолетовый;
- если смешать красный и желтый, то получится оранжевый;
- если смешать синий и желтый, то получится зеленый.

Напишите программу, которая считывает названия двух основных цветов для смешивания. Если пользователь вводит что-нибудь помимо названий «красный», «синий» или «желтый», то программа должна вывести сообщение об ошибке. В противном случае программа должна вывести название вторичного цвета, который получится в результате.

```
a1 = input()
a2 = input()
```

```
if a1 == 'красный':
    if a2 == 'красный':
        print('красный')
    elif a2 == 'желтый':
        print('оранжевый')
    elif a2 == 'синий':
        print('фиолетовый')
elif a1 == 'желтый':
    if a2 == 'красный':
        print('оранжевый')
    elif a2 == 'желтый':
        print('желтый')
    elif a2 == 'синий':
        print('зеленый')
elif a1 == 'синий':
    if a2 == 'красный':
        print('фиолетовый')
    elif a2 == 'желтый':
        print('зеленый')
    elif a2 == 'синий':
        print('синий')
if a1 not in ('синий', 'красный', 'желтый') or a2 not in ('синий', 'красный', 'желтый'):
    print('ошибка цвета')
```

Результат:
красный
синий
фиолетовый