

## Дополнительные операции

Мы познакомились с 4 основными математическими операциями в языке Python: сложение ( `+` ), вычитание ( `-` ), умножение ( `*` ) и деление ( `/` ). Добавив ещё 3 операции, мы получим инструментарий, достаточный для написания 99% программ.

Оператор	Описание
<code>**</code>	Возведение в степень
<code>%</code>	Остаток от деления
<code>//</code>	Целочисленное деление

## Возведение в степень

Оператор возведения в степень `a ** n` возводит число  $a$  в степень  $n$ . Рассмотрим работу данного оператора на примерах:

```
print(2 ** 0)
print(2 ** 1)
print(2 ** 2)
print(2 ** 3)
print(2 ** (-1))
```

Результатом выполнения такой программы будет:

```
1
2
4
8
0.5
```

**Обратите внимание:** оператор возведения в степень `**` может возводить не только в положительную степень, но и в отрицательную. Аналогично, основание степени  $a$  также может быть числом отрицательным.



**Запомни:** в отличие от многих языков программирования, в языке Python для возведения в степень используется оператор `**`, а не `^`.

**Обратите внимание:** оператор возведения в степень (`**`) является правоассоциативным (значение выражения вычисляется справа налево) в соответствии с правилами математики. Таким образом, выражение `x ** y ** z` вычисляется как `x ** (y ** z)`.

Результатом работы следующей программы:

```
print(2 ** 2 ** 3)      # 2 ** (2 ** 3) = 2 ** 8
```

будет:

256

## Целочисленное деление

Для положительных чисел оператор целочисленного деления ведёт себя как обычное деление, за исключением того, что он отбрасывает десятичную часть результата. Рассмотрим работу данного оператора на примерах:

```
print(10 // 3)
print(10 // 4)
print(10 // 5)
print(10 // 6)
print(10 // 12)
```

Результатом выполнения такой программы будет:

```
3
2
2
1
0
```

При делении отрицательных чисел необходимо помнить, что результат целочисленного деления не превосходит частное. Другими словами, округление берётся в меньшую сторону (число  $-4$  меньше, чем число  $-3$ ).

Результатом работы следующей программы:

```
print(10 // 3)
print(-10 // 3)
```

будут числа:

```
3    # округление в меньшую сторону
-4   # округление в меньшую сторону
```

## Деление с остатком

Оператор деления с остатком возвращает остаток от деления двух целых чисел. Рассмотрим работу данного оператора на примерах:

```
print(10 % 3)
print(10 % 4)
print(10 % 5)
print(10 % 6)
print(10 % 12)
print(10 % 20)
```

Результатом выполнения такой программы будет:

```
1
2
0
4
10
10
```



**Запомни:** при нахождении остатка от деления на целое число  $n$  может получаться результат  $0, 1, 2, \dots, n - 1$ . Например, при делении на 2 возможны остатки 0, 1, при делении на 3 возможны остатки 0, 1, 2 и т.д.

## Заполните пропуски

$23 // 7 =$

$20 // 5 =$

$2 // 5 =$

$123 // 10 =$

$-123 // 10 =$

## Заполните пропуски

$23 \% 7 =$

$20 \% 5 =$

$2 \% 5 =$

$123 \% 10 =$

Что будет выведено на экран в результате выполнения следующей программы?

```
a = 15 // (16 % 7)
b = 34 % a * 5 - 29 % 5 * 2
print(a + b)
```

Что будет выведено на экран в результате выполнения следующей программы?

```
a = 82 // 3 ** 2 % 7
print(a)
```



# Расстояние в метрах

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

## Формат входных данных

На вход программе подаётся натуральное число – количество сантиметров.

## Формат выходных данных

Программа должна вывести одно число – полное число метров.

**Sample Input 1:**

345

---

**Sample Output 1:**

3

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

# Мандарины

$n$  школьников делят  $k$  мандаринов поровну, неделящийся остаток остается в корзине. Сколько целых мандаринов достанется каждому школьнику? Сколько целых мандаринов останется в корзине?

## Формат входных данных

На вход программе подаётся два целых числа: количество школьников и количество мандаринов, каждое на отдельной строке.

## Формат выходных данных

Программа должна вывести два числа: количество мандаринов, которое достанется каждому школьнику, и количество мандаринов, которое останется в корзине, каждое на отдельной строке.

### Sample Input 1:

3  
6

---

### Sample Output 1:

2  
0

### Sample Input 2:

12  
6

---

### Sample Output 2:

0  
6

```
pupils = int(input())  
mandarins = int(input())  
a = mandarins // pupils  
b = mandarins % pupils  
print(a)  
print(b)
```

# Пересчет временного интервала

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

## Sample Input 1:

150

---

## Sample Output 1:

150 мин - это 2 час 30 минут.

---

## Sample Input 2:

50

---

## Sample Output 2:

50 мин - это 0 час 50 минут.

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