

Вычисления

1. Целочисленная арифметика

Для целых чисел определены операции `+`, `-`, `*` и `**`. Операция деления `/` для целых чисел возвращает вещественное число (значение типа `float`). Также функция возведения в степень возвращает значение типа `float`, если показатель степени — отрицательное число.

Но есть и специальная операция целочисленного деления, выполняющегося с отбрасыванием дробной части, которая обозначается `//` (она соответствует операции `div` в Паскале). Она возвращает целое число: целую часть частного. Другая близкая ей операция — это операция взятия остатка от деления, обозначаемая `%` (она соответствует операции `mod` в Паскале). Например:

```
print(17 / 3)    # выведет 5.66666666667
print(17 // 3)   # выведет 5
print(17 % 3)    # выведет 2
```

2. Действительные числа

В этом разделе речь пойдет о действительных числах, имеющих тип `float`.

Обратите внимание, что если вы хотите считать с клавиатуры действительное число, то результат, возвращаемый функцией `input()` необходимо преобразовывать к типу `float`:

```
x = float(input())
print(x)
```

Действительные (вещественные) числа представляются в виде чисел с десятичной точкой (а не запятой, как принято при записи десятичных дробей в русских текстах). Для записи очень больших или очень маленьких по модулю чисел используется так называемая запись «с плавающей точкой» (также называемая «научная» запись). В этом случае число представляется в виде некоторой десятичной дроби, называемой *мантиссой*, умноженной на целочисленную степень десяти (*порядок*). Например, расстояние от Земли до Солнца равно $1.496 \cdot 10^{11}$, а масса молекулы воды $2.99 \cdot 10^{-23}$.

Числа с плавающей точкой в программах на языке Питон, а также при вводе и выводе записываются так: сначала пишется мантисса, затем пишется буква `e`, затем пишется порядок. Пробелы внутри этой записи не ставятся. Например, указанные выше константы можно записать в виде `1.496e11` и `2.99e-23`. Перед самым числом также может стоять знак минус.

Напомним, что результатом операции деления `/` всегда является действительное число (`float`), в то время как результатом операции `//` является целое число (`int`).

Преобразование действительных чисел к целому производится с округлением в сторону нуля, то есть `int(1.7) == 1`, `int(-1.7) == -1`.

3. Библиотека math

Для проведения вычислений с действительными числами язык Питон содержит много дополнительных функций, собранных в библиотеку (модуль), которая называется `math`.

Для использования этих функций в начале программы необходимо подключить математическую библиотеку, что делается командой

```
import math
```

Например, пусть мы хотим округлять вещественные числа до ближайшего целого числа *вверх*. Соответствующая функция `ceil` от одного аргумента вызывается, например, так: `math.ceil(x)` (то есть явно указывается, что из модуля `math` используется функция `ceil`). Вместо числа `x` может быть любое число, переменная или выражение. Функция возвращает значение, которое можно вывести на экран, присвоить другой переменной или использовать в выражении:

```
import math

x = math.ceil(4.2)
y = math.ceil(4.8)
print(x)
print(y)
```

Другой способ использовать функции из библиотеки `math`, при котором не нужно будет при каждом использовании функции из модуля `math` указывать название этого модуля, выглядит так:

```
from math import ceil

x = 7 / 2
y = ceil(x)
print(y)
```

или так:

```
from math import *

x = 7 / 2
y = ceil(x)
print(y)
```

Ниже приведен список основных функций модуля `math`. Более подробное описание этих функций можно найти на [сайте с документацией языка Питон](#).

Некоторые из перечисленных функций (`int`, `round`, `abs`) являются стандартными и не требуют подключения модуля `math` для использования.

Функция	Описание
Округление	
<code>int(x)</code>	Округляет число в сторону нуля. Это стандартная функция, для ее использования не нужно подключать модуль <code>math</code> .
<code>round(x)</code>	Округляет число до ближайшего целого. Если дробная часть числа равна 0.5, то число округляется до ближайшего четного числа.
<code>round(x, n)</code>	Округляет число <code>x</code> до <code>n</code> знаков после точки. Это стандартная функция, для ее использования не нужно подключать модуль <code>math</code> .
<code>floor(x)</code>	Округляет число вниз («пол»), при этом <code>floor(1.5) == 1</code> , <code>floor(-1.5) == -2</code>
<code>ceil(x)</code>	Округляет число вверх («потолок»), при этом <code>ceil(1.5) == 2</code> , <code>ceil(-1.5) == -1</code>
<code>abs(x)</code>	Модуль (абсолютная величина). Это — стандартная функция.
Корни, логарифмы	
<code>sqrt(x)</code>	Квадратный корень. Использование: <code>sqrt(x)</code>
<code>log(x)</code>	Натуральный логарифм. При вызове в виде <code>log(x, b)</code> возвращает логарифм по основанию <code>b</code> .
<code>e</code>	Основание натуральных логарифмов $e = 2,71828...$
Тригонометрия	
<code>sin(x)</code>	Синус угла, задаваемого в радианах
<code>cos(x)</code>	Косинус угла, задаваемого в радианах
<code>tan(x)</code>	Тангенс угла, задаваемого в радианах
<code>asin(x)</code>	Арксинус, возвращает значение в радианах
<code>acos(x)</code>	Арккосинус, возвращает значение в радианах
<code>atan(x)</code>	Арктангенс, возвращает значение в радианах
<code>atan2(y, x)</code>	Полярный угол (в радианах) точки с координатами (x, y).
<code>degrees(x)</code>	Преобразует угол, заданный в радианах, в градусы.
<code>radians(x)</code>	Преобразует угол, заданный в градусах, в радианы.
<code>pi</code>	Константа $\pi = 3.1415...$