§16. Операции. Функции. Выражения

Запись арифметических выражений в языке Python

Вы уже знакомы с числовыми типами данных int и float и строковым типом str. К ним применимы различные операции и многочисленные функции.

Для **основных арифметических операций** в языке Python используются следующие обозначения:

+ сложение	- вычитание
* умножение	/ обычное деление
// целочисленное деление	% определение остатка от деления
** возведение в степень	

Допустим нам необходимо **вычислить выражение**, записанное в математическом виде таким образом:

$$\frac{2 * 17{,}56^2}{7 * 2{,}47 * 0{,}43}$$

Прежде чем написать строку, которая позволит подсчитать нам результат, сформулируем правила записи алгебраических выражений на языке программирования:

1. Выражения содержат числа, имена других переменных, знаки операций, круглые скобки, имена функций.
2. Разделителем целой и дробной части является точка.
3. Выражение записывается в одну строку (линейная запись выражений), символы последовательно выстраиваются друг за другом, проставляются ВСЕ знаки операций, используются круглые скобки.
4. Учитывается **приоритет** операций (порядок действий), для изменения которого используются круглые скобки. Действия выполняются в следующем порядке:
    1) действия в скобках;
    2) возведение в степень;
    3) умножение и деление, слева направо;
    4) сложение и вычитание, слева направо.

Таким образом, умножение и деление имеют одинаковый приоритет, более высокий, чем сложение и вычитание.

Следуя правилам записи арифметических выражений, мы должны перевести данную (математическую запись) дроби в линейную запись, то есть записать дробь в одну строчку. Так как и числитель, и знаменатель у нас сложные (то есть содержат два и более множителя), то при записи в линейную форму необходимо выражения в числителе и знаменателе взять в скобки. Таким образом, линейная запись такого выражения будет выглядеть следующим образом:

```
(2*17.56*17.56)/(7*2.47*0.43)
```

Если в выражении операторы имеют одинаковый приоритет, то они выполняются в направлении слева направо за исключением оператора возведения в степень (**), который выполняется справа налево.

Определим порядок действий компьютера при вычислении выражения:

```
c + b – 1 / 2 * 5
```

Первым будет выполняться деление, ½, затем умножение полученного числа на 5. После этого сложение c + b и, наконец, из полученной суммы вычитается результат ранее выполненного умножения.

Целочисленная арифметика.

Остановимся более подробно на операциях деления.

Операция / - обычное деление. Ответ может быть дробным (тип float). Например, 9 / 4 = 2,25. Такой ответ можно получить при привычном нам делении на калькуляторе.

Операция // (во многих языках программирования обозначается словом **div)** - целочисленное деление. При таком делении дробная часть отбрасывается (ответ получается типа integer).

Например, 9 // 4 = 2

Операция % (во многих языках программирования обозначается словом **mod)** вычисляет остаток от деления (ответ получается типа integer). Например, 9 % 4 = 1

Чтобы лучше понять, вспомним как делили в начальной школе, когда не знали о дробях и сравним с вычислениями выше. 9 : 4 = 2 (ост. 1)

В Python операция вычисления остатка выполняется по математическим правилам, то есть, как принято считать в Теории Чисел, остаток - это неотрицательное число. **Знак остатка совпадает со знаком делителя.**

Пример

```
c = 10 // 3    # Ответ: c = 3
d = 10 % 3     # Ответ: d = 1
e = -7 // 4    # Ответ: e = -2
f = -7 % 4     # Ответ: f = 1
```

Значения переменных e и f получились такими, потому что -7 = (-2*4)+1

Функции и модули

К величинам в Python применимы различные операции и многочисленные функции, некоторые из них приведены в таблице.

Функция	**Назначение**	**Тип аргумента**	**Тип результата**
abs(x)	Абсолютная величина (модуль) числа x	int, float	Такой же, как у аргумента
round(x)	Округление вещественного X до заданного количества знаков после запятой (по умолчанию - до нуля знаков, т. е. до ближайшего целого)	float	int, float
int(x)	Преобразование вещественного или строкового X к целому	str, float	int
sqrt(x)	Квадратный корень из X	int, float	float
pow(x,y)	Вычисляет x в степени y	int, float	int, float
sin(x)	Синус угла X, заданного в радианах	int, float	float
cos(x)	Косинус угла X, заданного в радианах	int, float	float
tan(x)	Тангенс угла X, заданного в радианах	int, float	float
random()	Случайное число от 0 до 1	-	float
randint(a, b)	Случайное целое число n из промежутка от a до b включительно	int	int

Надо запомнить! Аргумент функции всегда записывается в скобках.

На языке Python написано так много самых разных функций, что встраивать весь этот объем кода в сам язык нецелесообразно. Проблема доступа к дополнительным возможностям языка, обеспечиваемым функциями, решается с помощью модулей. Каждый модуль содержит набор функций, предназначенных для решения задач из определенной области.

Так, первые три из представленных в таблице функций встроены в язык Python. Чтобы их вызвать, не надо выполнять никаких дополнительных действий.

Что касается функций pow(x,y), sqrt(x), sin(x), cos(x), tan(x), то для их вызова предварительно надо подключить модуль math, в котором собраны математические функции. Две последние из приведённых в таблице функций требуют подключения модуля random, позволяющего генерировать случайные числа. Модуль graph нужен для рисования геометрических фигур, и т. д.

Для доступа к функциям модуля его надо импортировать в программу. После импорта интерпретатор будет «знать» о существовании дополнительных функций и позволит ими пользоваться. Подключение модуля осуществляется командой import. Например, команда from math import* подключает к программе все функции (так как стоит знак *) модуля math. После этого к ним можно будет обращаться так же, как к встроенным функциям:

```
y = sqrt(x)        z = sin(x)
```

Для того чтобы записать в переменную *a* случайное число в диапазоне от 1 до 10, можно использовать следующие операторы:

```
from random import *      # Подключаем модуль random со всеми функциями
a = randint (1, 10)       # Обращаемся к функции randint () как к встроенной
```

Контрольные вопросы и задания

1. Каков приоритет операций при вычислении значений выражений?
2. Какие операции обозначаются символами / // % ?
3. Из таблицы в параграфе перепишите и выучите функции и их назначение.
4. Как подключить модули *math, random, graph* со всеми их функциями? Запишите три соответствующие команды с пояснениями, какая из них что делает.
5. Как сгенерировать случайное целое число из промежутка от 1 до 5? Запишите соответствующую команду.