Выражения

Выражения состоят из ***операций*** и ***операндов***.

В роли операндов могут быть:

- литералы (здесь это пять);
- переменные (это *a* и *b*);
- функции (здесь это функция *abs*, параметром которой является переменная *x*; функция возвращает абсолютное значение, т. е. модуль *x*).

Литералом называется значение, которое воспринимается буквально, а не является именем какого-либо объекта.

Операциями в данном выражении являются операции умножения (звездочка), сложения (плюс) и деления (слэш).

Возможные арифметические операции в *Питоне*:

- ****** возведение в степень
- ***** умножение
- **/** деление
- **//** целочисленное деление
- **%** остаток от целочисленного деления
- **+** сложение
- **–** вычитание.

Операции расположены в порядке убывания приоритета операций (операции умножения и три операции деления обладают одинаковым приоритетом, сложение и вычитание тоже обладают одинаковым приоритетом). Для повышения приоритета операций используются круглые скобки.

Операции сравнения в *Питоне*:

- **==** равно
- **!=** неравно
- **>** больше
- **<** меньше
- **>=** больше или равно
- **<=** меньше или равно

Оператор присваивания

Оператор присваивания состоит из имени объекта, знака равенства и значения объекта (значение может быть результатом вычисления выражения).

```
имя = выражение
```

Примеры операторов присваивания:

```
a = 25
x=0
b="Привет"
c='Hello'
d = (a>=0)
k = 2>3
x=x+5
x1 = 1.34+5*x
t = b**2 – 4*a*c
z = (x/2+13)/(2*a)
```

Особый интерес представляют логические переменные *d* и *k*. Переменная *d* примет значение *True* (истина) – потому что результат сравнения переменной *a* с нулем будет истинным. Переменная *k* примет значение *False* (ложь) так как 2 не больше 3.

Пример простой программы

Попробуйте набрать следующую программу

```
num = int(input('Введите целое число: '))
digit = abs(num)%10
print('Последняя цифра числа равна',digit)
```

Первая строка – ввод целого числа *num*. Вторая строка – переменной *digit* присвоили остаток от деления модуля введенного числа на 10. Третья строка – вывод содержимого переменной *digit*.

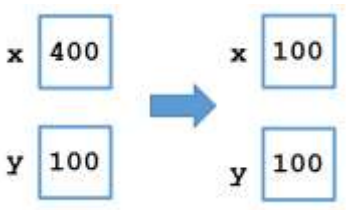
Дополнительная информация

Использование памяти в *Питоне*

Рассмотрим особенность размещения переменных в памяти в *Питоне*. В языках программирования таких как Си или Паскаль при выполнении программы

```
x = 400
y = 100
x = x-300
```

будут созданы два контейнера в памяти *x* и *y*, а затем будет изменено значение в контейнере *x*.



В *Питоне* же сначала создается в памяти объект, состоящий из трех частей: типа данных, значения и количества ссылок на этот объект.

Поэтому сначала создастся объект 400 с количеством ссылок равным нулю, затем переменная *x* сопоставится с адресом этого объекта, и количество ссылок станет равно одному. Подобные действия произойдут и с объектом 100, и переменной *y*.

Затем будет вычислено выражение 400–300=100. Так как уже есть объект с таким значением – объект, соответствующий имени *y*, то для имени *x* будет создано соответствие с этим же объектом. Количество ссылок на этот объект возрастет до двух, а количество ссылок на объект 400 станет равным нулю. Объект с нулевым количеством ссылок будет удален.