Для записи констант или при вводе-выводе может использоваться как привычное представление в виде десятично дроби, например 123.456, так и "инженерная" запись числа, где мантисса записывается в виде вещественного числа с одной цифрой до точки и некоторым количеством цифр после точки, затем следует буква ''e'' (или ''E'') и экспонента. Число 123.456 в инженерной записи будет выглядеть как 1.23456e2, что означает, что 1.23456 нужно умножить на 10**2. И мантисса и экспонента могут быть отрицательными и записываются в десятичной системе.

Такая запись чисел может применяться при создании вещественных констант, а также при вводе и выводе. Инженерная запись удобна для хранения очень больших или очень маленьких чисел, чтобы не считать количество нулей в начале или конце числа.

Если хочется вывести число не в инженерной записи, а с фиксированным количеством знаков после точки, то следует воспользоваться методом format, который имеет массу возможностей. Нам нужен только вывод фиксированного количества знаков, поэтому воспользуемся готовым рецептом для вывода 25 знаков после десятичной точки у числа 0.1:

```
1  x = 0.1
2  print('{0:.25f}'.format(x))
```

Вывод такой программы будет выглядеть как 0.1000000000000000055511151, что еще раз подтверждает мысль о том, что число 0.1 невозможно сохранить точно.

Проблемы вещественных чисел

Рассмотрим простой пример:

```
1  if 0.1 + 0.2 == 0.3:
2      print('Yes')
3  else:
4      print('No')
```

Если запустить эту программу, то можно легко убедиться в том, что 0.1 + 0.2 не равно 0.3. Хотя можно было надеятся, что несмотря на неточное представление, оно окажется одинаково неточным для всех чисел.

Поэтому при использовании вещественных чисел нужно следовать нескольким простым правилам:

1) Если можно обойтись без использования вещественных чисел - нужно это сделать. Вещественные числа проблемные, неточные и медленные.

2) Две вещественных числа равны между собой, если они отличаются не более чем на epsilon. Число X меньше числа Y, если X < Y - epsilon.

Код для сравнения двух чисел, заданных с точностью 6 знаков после точки, выглядит так:

```
x = float(input())
y = float(input())
epsilon = 5 * 10**-7
if abs(x - y) < 2 * epsilon:
    print('Equal')
else:
    print('Not equal')
```

В случае, если над числами совершались какие-то действия, то значения epsilon нужно вычислять как в приведенном в первом видео примере. В учебных задачах это можно сделать не внутри программы, а один раз руками для худшего случая и применять вычисленное значение как константу.

Пометить как выполненное