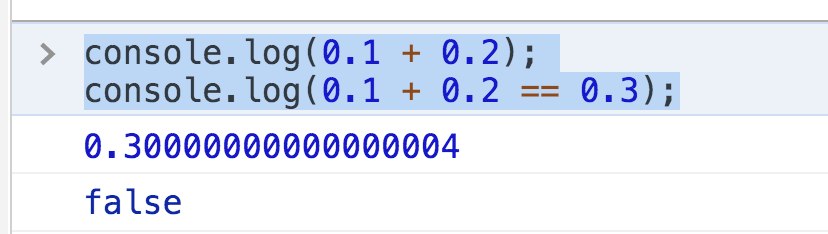
**Exercise1**

объяснить откуда там нули в первой строке

почему они не равный на второй строке



Вещественных чисел существует бесконечно много, но формат их представления в JS позволяет точно выразить лишь их ограниченное количество, это значит, что при работе с вещественными числами в JS представление числа часто будет являться округлением фактического числа.

**Exercise2**



в первой строчке js не считает, что “2” и “2” – это числа и просто складывает их между собой в строчку, прибавляя до этого к ним число 1, которое является для js числом.

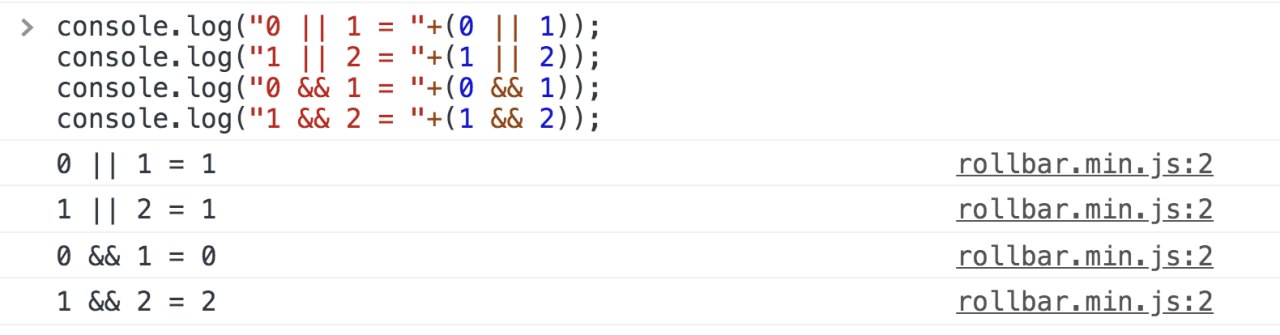
Во второй строчке + перед первой двойкой дает понять js, что это число и складывает с ним единицу и получая в ответе тройку и прибавляя к ней последнюю двойку, которая для js просто символ.

Такая же история с третий строчкой, только первым действием из цифры один вычитается цифра 1, получается 0, к которому прибавляется символ 2.

В четвертое строке такай же история, как и с первой строкой.

Пятая и шестая строка: дает NaN2 и NaN, так как NaN – это результат любого бессмысленного вычисления, но в первом случае 2ойка в кавычках считается строкой и перед ней стоит +, который по правилам js прибавляет ее к NaN.

**Exercise3**

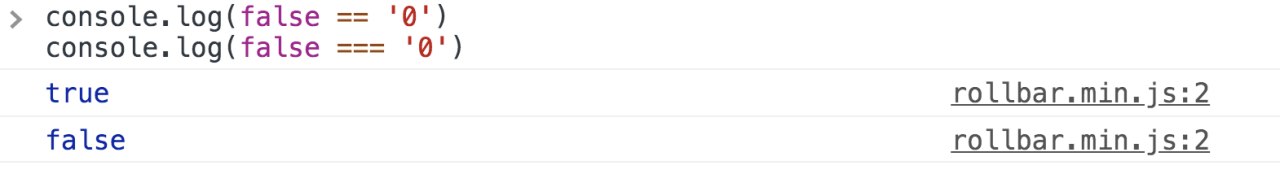


В первом случае оператор || возвращает первое истинное значение идя слева направо, так как 0 – это false в первом случае выводится 1.

Во втором случае оператор || сразу находит истинное значение и сразу его возвращает.

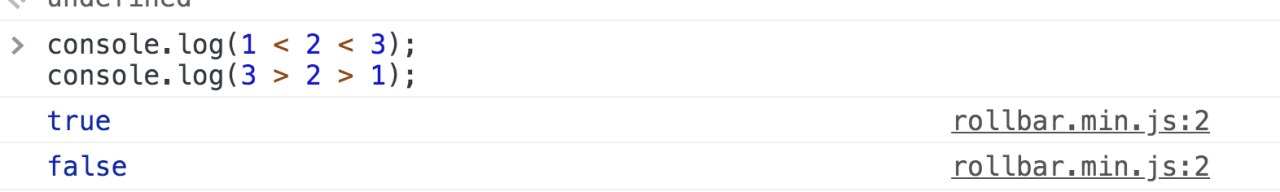
В третьем случае оператор && возвращает первое ложное значение – это 0 или последнее значение если не найдено ложных, как в четвертом случае

**Exercicise4**



В первом случае – приведение типов и не строгое сравнение, а во втором - строгое

**Exercise5**



JS интерпретирует слева направо, поэтому в первом выражение получается true<3, а во втором false>1 (и как я понял тру = 1, а false =0)

**Exercise6**

Из целочисленного числа сделать числи с плав точкой

let number = 3

alert (number+0.1);