* Какое будет выведено значение: let x = 5; alert( x++ ); ?

Выведется 5 потому, что единицу прибавит уже после вывода.

* Чему равно такое выражение: [ ] + false - null + true ?

Выводится NaN так как происходит попытка выполнить вычитание. Если бы его не было, то всё выражение просто будет преобразовано и объединено в строку

* Что выведет этот код: let y = 1; let x = y = 2; alert(x); ?

х не принял первое значение у потому, что присвоение в данном случае выполняется справа налево"

* Чему равна сумма [ ] + 1 + 2?

Выводится 12 потому, что пустой массив хранит в себе строчную пустоту=) и поэтому все остальные элементы преобразуются в строчные)

* Что выведет этот код: alert( "1"[0] )?

Данное действие выводит первый элемент строки, в данном случае это 1

* Чему равно 2 && 1 && null && 0 && undefined ?

Логический оператор И останавливается при первом ложном значении и выводит его, в данном случае это null

* Есть ли разница между выражениями? !!( a && b ) и (a && b)?

В первом случае логический оператор Не выполняет преобразование в булевый тип, а именно в False, потом опять в true. А во втором случае выражение истинно и поэтому выводится последнее правдивое значение

* Что выведет этот код: alert( null || 2 && 3 || 4 ); ?

Сначала выполняется И, которое выводит 3, потом ИЛИ, которое видит 3 - истину, а значит останавливает дальнейшее выполнение"

* a = [1, 2, 3]; b = [1, 2, 3]; Правда ли что a == b ?

Выводится ложь, но я не знаю почему=), скорее всего ихха того, что сравниваются просто имена массивов

* Что выведет этот код: alert( +"Infinity" ); ?

В данном случае происходит явное преобразование строки в число и выведется Infinity в числовом типе.

* Верно ли сравнение: "ёжик" > "яблоко"?

По юникоду кодировка ё больше чем я, поэтому выражение правдиво)

* Чему равно 0 || "" || 2 || undefined || true || falsе ?

Здесь все значения выводят ложь, кроме true, и так как используется логический оператор ИЛИ, то нам достаточно, чтобы только один элемент был правдой и именно первый такой элемент и выведется