**Вопросы к этому заданию**

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

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

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

**Чему равна сумма [ ] + 1 + 2?**

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

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

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

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

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

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

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

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

#### Пример преподавателя

Фото пользователя

[Иван Петриченко](https://www.udemy.com/user/yan-kovalenko-2/)

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

Выведет 5. Это задача на разницу между постфиксной и префиксной формой инкремента/ декремента.

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

NaN - Not a number.

Пустой массив преобразуется в пустую строку. - В 18 лекции была речь о том, что пустой массив, как булиновое значение - это true. Да, я помню, что в этой же лекции говорилось о сложении чего угодно с пустой строкой - и в итоге мы получаем строку.

Почему в этом случае пустой массив принимается кодом именно как пустая строка, а не как булиновое значение?

Далее идет сложение пустой строки и false - получается "false"

Когда мы пытаемся от строки отнять что-то - выполняется математически неверная операция и мы получаем NaN

Это задача на динамическую типизацию в JS (по простому - преобразование типов данных один в другой)

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

Выведет 2.

В этой части y = 2 мы присваиваем переменной y значение 2, а затем это же значение идет в переменную  x

**Чему равна сумма [ ] + 1 + 2?**

Выведет "12"

Пустой массив превращается в пустую строку, а дальше при сложении числа со строкой - все преобразуется в тип данных строка.

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

Выведет 1, так как это первый символ в этой строке.

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

Выведет null, так как оператор И возвращает первое ложное значение.

В логическом контексте в JS только 5 сущностей представляют собой неправду (false) - это 0, пустая строка "" , null , undefined , NaN

мы понимаем, что в булиновом контексте null - это false

Так почему возвращается именно null, а не false?

Или это такая аксиома - возвращается суть первой "лжи", о которую запнулся оператор &&?

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

Есть. !!  превращает любую конструкцию в Boolean тип данных

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

Выведет 3. Почему так? Мы разбиваем этот кусочек кода на 3 части.

Сначала, в операции ИЛИ - из двух частей вернет 2. ИЛИ возвращает всегда первое правдивое утверждение.

После оператор И сравнит 2 и 3 и вернет последнее - 3ку. Оператор И возвращает первое ложное значение, но если его нет - то последнее.

И в конце ИЛИ возвращает первое правдивое утверждение.

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

Нет. Это разные массивы, просто с одинаковыми значениями.

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

Вернет Infinity. Оператор + перед строкой превратит её в число.

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

Нет, так как в соответствии с unicode кодировкой ё стоит после я.

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

Вернет 2, так как это первое правдивое значение (true)