Для понимания работы логических операторов, прочитать

<https://learn.javascript.ru/logical-ops>

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

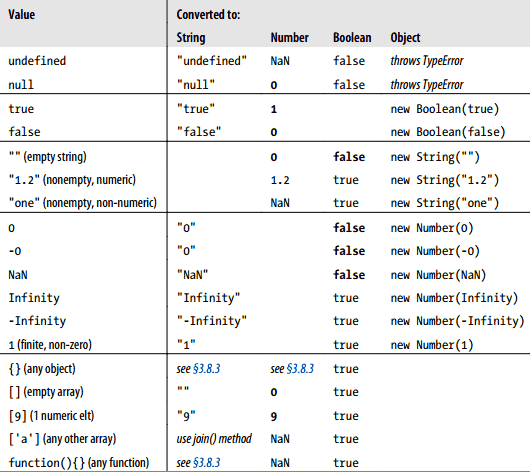
*Ответ: 5*

Разбор был в группе Академии: <https://vk.com/glo_academy?w=wall-123083697_20188>

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

*Ответ: NaN*

Здесь нужно знать, как происходит преобразование типов в JS.   
Почитать дополнительно об этом можно здесь: <https://learn.javascript.ru/types-conversion>. Общие правила представлены в таблице:



Начнем складывать по порядку. [ ] + false не являются числами, поэтому они будут складываться как строки, то есть получится "false". Если бы не было вычитания – null, то дальше выражения сложились бы как строки, то есть получилась бы строка "falsetrue".

Но дальше нам приходится иметь дело с вычитанием, а вычитание строк в JS не предусмотрено, эта операция выполняется с приведением строк к числу. И если null может быть преобразован к числу (см.таблицу – получился 0), то строка с нечисловым значением внутри не может быть приведена к числу (см. таблицу – строка "false" преобразуется к NaN). Получается, NaN - null = NaN.

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

*Ответ: 2*

Здесь все просто, сначала переменной присваивается значение y = 1, потом переменной x присваивается значение y, а потом им обоим присваивается значение 2. Теперь и x равен 2, и y равен 2.

* **Чему равна сумма [ ] + 1 + 2?**

*Ответ: “12”*

Как и в примере выше, пустой массив при сложении преобразуется в строку, при сложении какого-либо литерала со строкой, они оба будут преобразованы в строку, поэтому все литералы по очереди преобразуются в строку, пустая строка + "1" + "2" равно строке “12”.

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

*Ответ: 1*

Здесь "1" является обычной строкой. Чтобы получить символ строки, мы можем воспользоваться вызовом charAt(позиция), а также можно использовать квадратные скобки. Первый символ строки имеет позицию 0. В нашем примере вызван первый и единственный символ строки с позицией 0.

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

*Ответ: null*

Разбор был в группе Академии: <https://vk.com/glo_academy?w=wall-123083697_16122>

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

*Ответ: да*

Разбор был в группе Академии: <https://vk.com/glo_academy?w=wall-123083697_21605>

Если простым языком: в случае с выражением a && b у нас вернется либо a, либо b, а в случае с !!( a && b) у нас вернется значение либо true, либо false.

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

*Ответ: 3*

Здесь сначала будет выполнено выражение 2 && 3, так как приоритет у && больше, чем у ||. Так как оператор && проводит вычисления до первого ложного значения и возвращает его, а если все истинные – то последнее значение, соответственно 2 && 3 вернет 3.

Далее, вычисляем null || 3 || 4. Так как оператор || возвращает первое найденное истинное значение, или, если таких не нашлось, то последнее ложное, то данное выражение возвращает первое истинное значение 3.

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

*Ответ: нет*

Разбор был в группе Академии: <https://vk.com/glo_academy?w=wall-123083697_19499>

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

*Ответ: Infinity*

Здесь все просто: оператор + перед строкой "Infinity" приводит ее к числу, а в JS существует специальное числовое значение Infinity (бесконечность), оно и будет выведено.

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

*Ответ: да*

Про сравнение строк и других типов можно почитать тут: <https://learn.javascript.ru/comparison>  
Сравнение строк идет по первому символу (и далее, если они равны). Сравниваются численные коды символов, в данном случае используется кодировка Unicode. Unicode для буквы "ё" идёт после буквы "я", т.е. имеет большее значение поэтому "ё" > "я" (u0451 > u044f), таким образом сравнение верно.

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

*Ответ: 2*

Как уже было сказано ранее, оператор || возвращает первое найденное истинное значение, в данном случае это 2.