1. Почему код дает такие результаты:

var a = 1, b = 1, c, d;  
c = ++a; alert(c);           *// 2*  - по порядку записи сначала «с» присваивается значение «++а», затем «а» увеличивается на 1, затем выводится «с», равное а+1, то есть 2  
d = b++; alert(d);           *// 1* - по порядку записи сначала «d» присваивается з «b», которое = 1, затем выводится значение на экран командой, а в самом конце «b» увеличивается на 1, поэтому в alert это не попадает.  
c = (2+ ++a); alert(c);      *// 5* – «а» уже увеличено на 1 в предыдущих строках кода, затем происходит присваивание переменной «с» выражения в скобках, в которых 2 прибавляется к префиксной ++а, при префиксной записи сначала происходит увеличение переменной «а» (2+1), затем прибавляется 2, что приводит к значению 5.  
d = (2+ b++); alert(d);      *// 4* – то же самое, что и в предыдущем коде, только в постфиксной форме записи сначала происходит сложение 2 и «b», что дает 4, затем выводится на экран переменная «d», затем «b» увеличивается на 1.  
alert(a);                    *// 3* – по результатам вышепреведенных действий переменная «а» дважды увеличивалась оператором ++  
alert(b);                    *// 3* - по результатам вышепреведенных действий переменная «b» дважды увеличивалась оператором ++

1. Чему будет равен «х»

var a = 2;  
var x = 1 + (a \*= 2);

х будет равен 5, так как а умножается на 2, затем прибавляется 1

1. \* Сравнить **null** и **0**. Объяснить результат.

Null > 0 false

Null < 0 false

Null == 0 false

Null >= 0 true

Nul <= 0 true

Принцип сравнения связан со спецификацией к javascript,

Дело в том, что в спецификации к оператору >= есть описание «Если null < 0 принимает значение false, то null >= 0 принимает значение true»