**1.** false && true || true; // true. Операция выполняется слева направо. Результатом первой операции будет false (&& возвращает true только, если оба аргумента истинны), далее получаем false || true, где получим true, поскольку при сравнении возвратится true, если одна из операнд true.

**2.**  false && (true || true); // false. Сначала выполнится операция в скобках с результатом true, затем получаем false && true, где результатом будет false.

**3.**  +'40' + +'2'; // 42. '40' и '2' числовые строки. Унарный плюс перед '40' преобразовал строку в число, поэтому, следующие 2 плюса - не инкремент и обозначают сложение и унарный плюс, преобразующий '2' в число. После преобразования строк получаем

40 + 2; //42

**4.** '2' + 3 \*\* 2; // '29'. Сначала 3 возводится во вторую степень (==9), потом к строке '2' прибавляется число 9 и в результате конкатенации получаем строку.

**5.**  3 \*\* 2 / 3; // 3. Действия совершаются последовательно, в том порядке, в котором они написаны.

**6.**  3 \*\* (9 / 3); // 27. Вначале выполняется деление в скобках, затем первый операнд возводится в степень, полученную в результате деления.

**7.**  '10' + (5 === 6); // '10false'. В скобках вследствие строгого сравнения 5 и 6 получаем false. А при сложении строки и false получаем преобразование булевого типа данных в строку и их конкатенацию.

**8.** '10' - 5 === 6; // false. '10' преобразуется в число. При вычитании получаем число 5, при строгом сравнении которого с числом 6 получаем false.

**9.**  undefined + 1; //NaN. Если undefined преобразовать в число, получаем NaN, а при любых математических действиях, если одна из операнд NaN, результат будет также Nan

**10.**  (+null == false) < 1; // false. Унарный плюс преобразует null в числовой тип данных (получаем 0). Результатом получившейся операции в скобках (0 == false) будет true и, следовательно, 1<1 == false.

**11.**  (+null == false) < 1 \*\* 5; // false. Все так же, как в предыдущем вопросе, поскольку 1 в степени 5 все равно 1.

**12.**  var number = 5;

2 \* ++number; //12. Префиксная форма инкремента, поэтому получаем 2 \* 6

2 \* number--; //10. Постфиксная форма декремента - 2 \* 5.

console.log(number); //5. Мы не присваивали переменной новых значений, поэтому консоль вернет изначальное значение переменной.