1. Prefix & postfix:

B= ++a

B1: +1 cho a

B2: tra ve a sau khi duoc +1

B = a++

B1: gan a cho B

B2: +1 cho a

B3: tra ve a copy

Vd: var number = 10;

Number = number++\*2 - --number\*2 = 10\*2 – 10\*2 = 0

1. String:

A = 1 + 2 + ‘3’ => A = ‘33’

B = ‘3’ + 2 + 1 => B = ‘321’

1. If

6 values mean FALSE: 0 - False - undefined - Null – NaN - ‘ ’/ “ “

Con lai la TRUE. VD: ‘0’ != 0

Vd: var a = 0;

If (a) {} => false moi chay

1. Primitive Data:

**Number:** a = 1, b = 2,…

**String:** a = ‘Hoang’ , “2”, ‘Nguyen \”Hoang” , ‘Nguyen \’Hoang’, “Nguyen ‘Hoang”,…

**Boolean:** isSuccess = true, isSuccess = false;

**Undefined:** var age; => khong gan gia tri

**Null:** var isNull = null; //nothing

**Symbol:** var id = **Symbol**(‘id’); // unique

1. Complex Data:

**Function:** var myFunction = **function**() {

Alert();

}

myFunction(); => call/goi

**Object:**

1. Object:

Var myObject = {

name: ‘Nguyen Hoang’,

age: 18,

address: ‘Vaasa’

// keys

myFunction: function() {

}  
}

1. Array:

Var myArray = [

‘Javascript’,

‘PHP’,

‘Python’

];

**CHECK TYPE OF DATA:** console.log(**typeof** a)

(voi null type thi check van ra object)

1. Comparison:

a == b / a != b : chi so sanh value

a === b / a !== b : so sanh value+type of data

1. Logical Arithmetic vs Comparison Arithmetic vs IF:

Var result;

Var a = 1; b = 2;

If (result) {

Alert(a);

} else {

Alert(b);

}

Case 1: result = a < b hoặc result = a > b

* Đều trả thẳng ra boolean: true/false , truyền vào if ()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Case 2: result = a && b

B1: xét từng vế xem có trùng cái nào trong “6 false-values” không

B2: nếu không có, **gán vế cuối cùng** cho ‘result’ ( result = b)

Hoặc: nếu trùng 1 trong 6 thì nó sẽ **gán giá trị sai đầu tiên** cho ‘result’

Vd: result = null && a;

B3: truyền vào if (), **giá trị là ‘true’**

Hoặc: truyền vào if (), **giá trị ‘false’**

**Ví dụ:** result = ‘A’ && ‘B’ && ‘C’

* Result = ‘C’ => if (result) = if (true)

result = null && ‘C’;

* Result = null => if (result) = if (false)

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Case 3: result = a || b

B1: xét từng vế xem có trùng cái nào trong “6 false-values” không

B2: nếu không phải tất cả đều là ‘6 false-values’, **gán vế đầu tiên** cho ‘result’ ( result = a)

Hoặc: nếu tất cả các vế đầu trùng 1 trong 6 thì nó sẽ **gán giá trị sai cuối cùng** cho ‘result’

Vd: result = null || undefined;

B3: truyền vào if (), **giá trị là ‘true’**

Hoặc: truyền vào if (), **giá trị ‘false’**

**Ví dụ:** result = ‘A’ || ‘B’ || ‘C’

* Result = ‘A’ => if (result) = if (true)

result = null || ‘C’ || ‘A’;

* Result = ‘C’ => if (result) = if (false)

Result = null || 0 || false;

* Result = false => if (result) = if (false)

**Comparison Arithmetic: TRẢ LẠI THẲNG GIÁ TRỊ BOOLEAN CHO BIẾN ĐC GÁN**

**Logical Arithmetic: trả giá trị theo 1 vế.**