[] == 0; //==========================================true

[] == false; //===============================================true

[] == undefined; //=====================================false

[] == null; //==========================================false

[] == ''; //==============================================true

0 == false; //=========================================true

0 == undefined; //=======================================false

0 == null; //==========================================false

0 == ''; //===========================================true

false == undefined; //===================================false

false == null; //====================================false

false == ''; // ========================================true

false == 'false'; //===================================fasle

undefined == null; //=====================================true

undefined == ''; // ======================================false

undefined == 'undefined'; //==============================false

null == ''; //==========================================false

# **两个不同类型数据比较时隐式转换规则**

　　1)  数字和字符串、布尔类型、数组进行比较时，字符串（或布尔类型、或数组）先转换为数字（Number），再进行比较；数组转数字请参看后文。

123 == '123asd'; //false

123 == '123'; //true

123 == false; //false

123 == true; //false

　　使用 if(num) 和 if(num == true) 结果不一样。前者使用Boolean()转换Num， 后者使用Number转换true。

[IMG_256](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

var num = 123;

if(num) {

console.log(1); //输出1 }

if(num == true) {

console.log(2); //无输出 }

var str = "qwe";

if(str) {

console.log(1); //输出1 }

if(str == true) {

console.log(2); //无输出

}

[IMG_257](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

　　2) 字符串和布尔类型比较，看完下面的代码我疯了。如果一定要给一个解释，那么一定是进行比较的两个数据同时经过Number()的洗礼后再进行比较。数组和布尔类型的比较也如此。

[IMG_258](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

"qwe" == false; //false

"qwe" == true; //false

"" == false; // true

"" == true; //false

[] == false; //true

[0] ==false; //true

[00] ==false; // true

[0,0] == false ; //false

[IMG_259](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

 　　3) undefined 除了和null进行非全等比较返回true,其它均返回false。null 除了和 undefined进行非全等比较返回true,其它均返回false。

　　4) 数组（或对象）和字符串进行比较时，数组（或对象）会转换成字符串再进行比较。

[IMG_260](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

[] == ""; //true

[0] == ""; //false

[0] == "0"; //true

var obj = {a: 1};

obj == ""; //false

obj == "[object Object]"; //true

[IMG_261](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

# **Number() 强制转换成数值**

　　1) 字符串转数值，能解析的返回数值，不能解析的返回NAN（Not a Number）。

Number('123'); //123

Number('123shj'); //NAN

parseInt('123shj'); // 123  
　　 Number('\t\v\r12.34\n'); // 12.34  
　　 parseInt('\t\v\r12.34\n'); // 12

　　2) 布尔值转数值

Number(true); // 1

Number(false); //0

　　3) Unfined 转数值

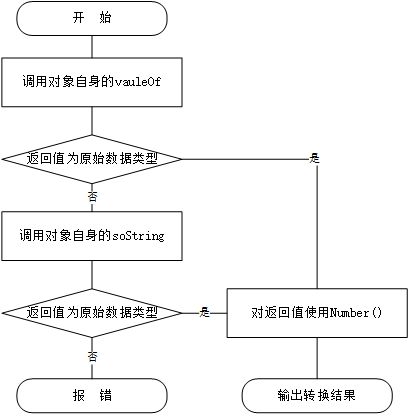
Number('undefined'); // NAN

　　4) null 转数值

Number(null); // 0

　　5) 对象转数值

Number({a: 2}); // NAN



# **String() 强制转换成字符串**

　　1) 原始数据类型转换为字符串，相当于直接把原数据用引号包起来。

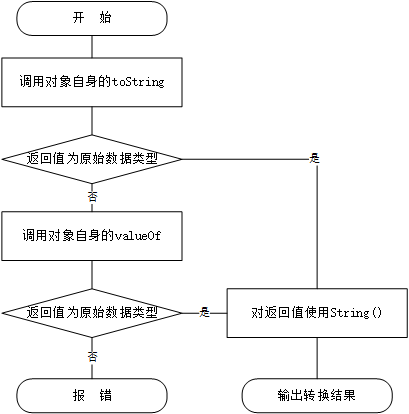
String(123); // "123"

String(true); // "true"

String(undefined) // "undefined"

String(null); // "null"

　　2) 对象转字符串



# **Boolean() 强制转换成布尔类型**

　　所有对象转换为Bool类型都为true，只有以下几个数值转换为Bool为false: undefined/null/0/NAN/""。

　　 undefined == false 返回 false， Boolean(undefined) = false.

　　 [] == false 返回true,  Boolean([]) = true.

[IMG_264](http://www.cnblogs.com/codelovers/archive/2016/04/01/javascript:void(0);)

Boolean(undefined); //false

Boolean(null); //false

Boolean(0); //false

Boolean(NAN); //false

Boolean(""); //false

Boolean([]); //true

Boolean({}); //true

Boolean(new Boolean(false)); // true