### typeof与instanceof与constructor

typeof(a) || typeof a可以判断a的类型，typeof可以返回的类型有：”number” “String” “Object” ”boolean””function” ”object” “undefined”。

值得注意的是，任何 自定义函数 和 数组 以及Null 都会被返回object。

typeof还可以访问不存在的东西，比如一个从未声明的变量。会返回”undefined”。

注意，一个从未声明的变量属于”undefined”而不是Null。

我们也可以使用typeof来结合if来判断某些值是否存在。或者这个值属于什么类型。

在这之前需要理解一下下面的代码：

<script>

*console*.log(typeof a); // a未声明，则为undefined，typeof可以检测不存在的东西

var b = *Boolean*(a); // a为声明，直接报错a is not defined

if(a){ // 如果注释掉上一行，则也会报错，毕竟if括号中其实就是执行了上一行的过程

alert("你好");

}

</script>

我们通常都会像这样使用if(a)来判断某个变量是否存在。但是如果a并没有被声明，那么这个程序就会出现致命错误，也就无法进行程序。这样的程序看上去理所当然，其实不然，这也是新手常犯的错误，bug相当难找。

然而声明过就不会出现这样的问题

<script>

var a = undefined; // 也可以是null

*console*.log(typeof a); // a未声明，则为undefined，typeof可以检测不存在的东西

var b = *Boolean*(a); // a为声明，直接报错a is not defined

if(a){ // 如果注释掉上一行，则也会报错，毕竟if括号中其实就是执行了上一行的过程

alert("你好");

} else {

alert("哈哈");

}

</script>



理解了typeof 我们就可以这样更好的判断

if(typeof a == "undefined"){ // a虽然未声明，但是依旧返回undefined,if的布尔判断正常

alert("你好");

} else {

alert("哈哈");

}

但是应该注意，如果a是一个null，typeof就会返回object。这样有时会影响我们的判断。

下面这个是一个错误的做法，因为a为undefined的时候也会返回相同结果，《Javascript高级程序设计》中曾着重讲到这个问题

尽管undefined和null是不同的东西，但是js在这儿判断的时候就是这么让人费解。

var a = undefined;

*console*.log(typeof a);

if(a == null ){ // a 为 undefined或者0

alert("你好"); // 弹出你好

} else {

alert("哈哈");

}

有些时候我们获取的dom得到的也是null，我觉得只需要这么判断就好

// 并不存在id为a的元素

*console*.log(document.getElementById("a")); // null

if(document.getElementById("a") == null){

*console*.log("不存在");

} else {

*console*.log("存在");

}

而不能想下面这么判断！这也是常见的不好找的致命错误！！

*console*.log(document.getElementById("a")); // null

if(document.getElementById("a").length == 0){

*console*.log("不存在");

} else {

*console*.log("存在");

}

因为null根本就没有length这个属性。

我们可以使用 typeof 来获取一个变量是否存在，如 if(typeof a!="undefined"){alert("ok")}，而不要去使用 if(a) 因为如果 a 不存在（未声明）则会出错，对于 Array,Null 等特殊对象使用 typeof 一律返回 object，这正是 typeof 的局限性。

经常会在js里用到数组,比如 多个名字相同的input, 若是动态生成的, 提交时就需要判断其是否是数组.   
  
　　　　　if(document.mylist.length != "undefined" ) {} //这个用法有误.   
  
正确的是 if( typeof(document.mylist.length) != "undefined" ) {}   
  
 或 if( !isNaN(document.mylist.length) ) {}   
  
typeof的运算数未定义,返回的就是 "undefined".   
  
**在 JavaScript 中，判断一个变量的类型尝尝会用 typeof 运算符，在使用 typeof 运算符时采用引用类型存储值会出现一个问题，无论引用的是什么类型的对象，它都返回 “object”。这就需要用到instanceof来检测某个对象是不是另一个对象的实例。**

https://www.cnblogs.com/Trr-984688199/p/6180040.html