

李姚组

1.js数据类型的分类(李姚)

2.js有那几种情况下为false

```
NaN,null,undefined,0,""
```

3.JS中使用typeof能得到那些类型(李姚)

```
1 typeof undefined // undefined
2 typeof 'abc' // string
3 typeof 123 // number
4 typeof true // boolean
5 typeof {} // object
6 typeof [] // object
7 typeof null //object
8 typeof console.log // function
9 typeof NaN // number
10 typeof class foo {} // function
11 typeof Symbol() // Symbol
```

7.`null`的值从技术上来说和`object`和`number`一样，都是最基本的值，按理来说，`null`的类型也应该是"null"。然而并非如此，因为JavaScript最初设计时出了一点意外。

在JavaScript最初设计时，一个值有两个部分组成：它的类型标签和实际的值。有5个类型标签可以使用，而且对象类型的引用指向 0。`null` 的值始终指向 NULL 指针，它在大部分平台都是用 0x00 来表示。由于这种相似性，`null` 就用过 0 类型标签来表示，所以符合对象的引用。

8. 因为函数是一个特殊的引用类型，在js中函数的定位很高，在任何情况下，都能typeof识别函数。

9. `NaN`代表某个值不是一个数字，但出乎意料的是，它是"number"类型。原因是这样的，在计算机内部，`NaN`是以数字类型储存的。然而，它是一个不能用实际数字来表示的数值类型的值。所以它叫"Not a Number"，这并不意味着它不是数值类型。相反，它意味着这个值不能用数值表示。

10. JavaScript的类只是一个被语法糖包裹的函数方法。实际上创建了一个同样的函数，但是作者的写法不同，只是看起来个简洁。这就是为什么`typeof`一个类，得到的仍然是"Function"。

11. es6新类型。`Symbol`是用来定义对象的唯一属性名的不二之选。

```
Symbol("foo") === Symbol("foo"); //输出: false
```

也就是说他自己都不等于自己

童建设组

徐保山组

程超
