### 深拷贝与浅拷贝

#### 1.数据类型

数据类型：（1）基本数据类型（String，Null，Boolean，Number和Undefined）

（2）引用数据类型（Object）

声明变量时不同的内存分配：

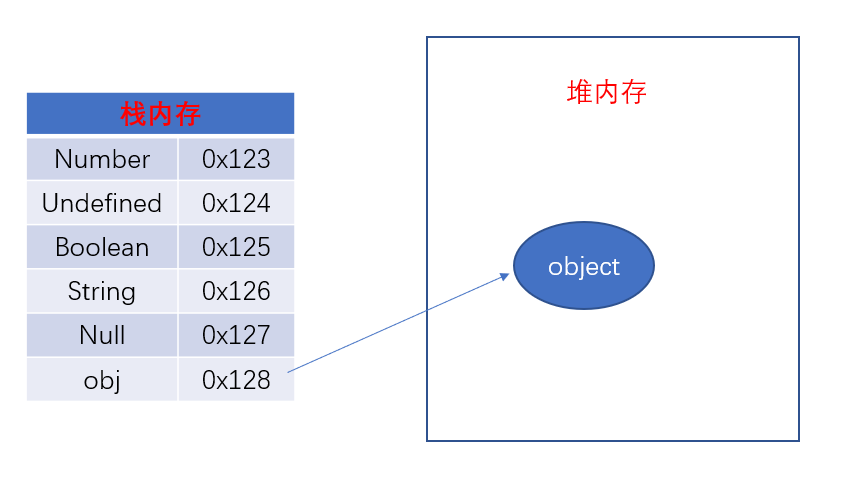
|  |  |
| --- | --- |
| 基本数据类型 | 数据较小并且不会改变，存放在栈内存中便于查找。 |
| 引用数据类型 | 会改变，放在堆内存中，栈内存中存放指向数据的指针 |

传递参数：

ES中所有函数的参数都是按值传递的。

访问变量可以有按值访问和按引用访问两种方式。

|  |  |
| --- | --- |
| 基本数据类型 | 把变量里的值传递给参数，之后参数和变量互不影响（两个独立个体） |
| 引用数据类型 | 对象变量里的值是堆内存中的内存地址，传递的值是这个内存地址（两个指针指向同一个对象） |



#### 2.深拷贝与浅拷贝

深拷贝：新建一个内存来存放拷贝的对象，独立的个体，复制后与原对象无关。

浅拷贝：使用原来对象的内存，只复制内存的引用地址，复制后与依赖原对象。

对于基本数据类型来说，只要是拷贝，只能是深拷贝。对于引用类型才区分深拷贝和浅拷贝。

|  |
| --- |
|  |
| 基本数据类型的拷贝，只是值相同，但是内存地址不同， |

|  |  |
| --- | --- |
|  |  |
| 浅拷贝 | 深拷贝 |

#### 3.赋值

当把一个对象赋值给一个新变量的话，无论是基本数据类型还是引用数据类型都是赋的内存地址，而不是创建新的空间，所以完全依赖原对象。

// 对象赋值

var obj1 = {

   'name' : 'zhangsan',

   'age' :  '18',

   'language' : [1,[2,3],[4,5]],

};

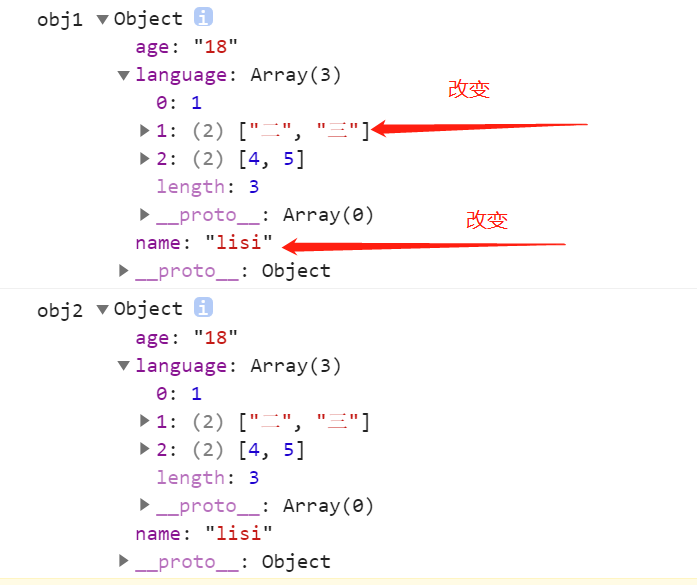
var obj2 = obj1;

obj2.name = "lisi";

obj2.language[1] = ["二","三"];

console.log('obj1',obj1)

console.log('obj2',obj2)



#### 4.拷贝的实现方式