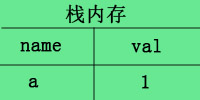
**1.基础数据类型：string number boolean null undefined symbol（es6新增）**

**引用（复杂）数据类型：array object 函数**

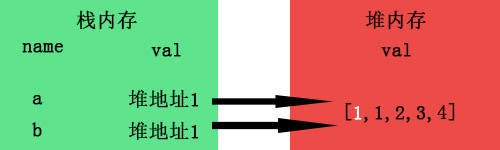
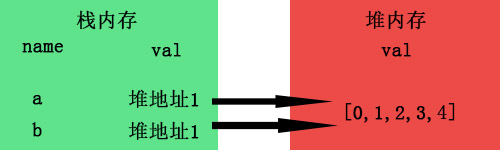
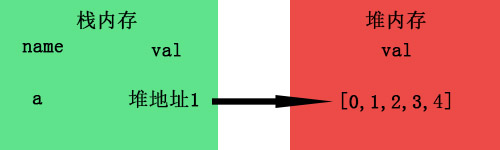
**PS：js的数据类型一般回答时只说有7种：null undefined number string boolean object symbol (可能array和函数归为特殊的object)**

**2.深拷贝只针对复杂数据类型：**

**a.基本类型--名值存储在栈内存中,浅拷贝的两个变量各自的值存放在自己的内存，所以修改值，相应的内存数据改变**

**b.引用类型—名存在占内存中，值存放在堆内存，并且栈内存会存放一个引用的地址指向堆内存的数据**

****

**所以，改变堆内存中的数据，由于变量都指向这个堆内存地址，所以两个变量的数据都随着改变；所以实现深拷贝的原理是各自有自己的堆内存存放数据，改变时互不影响**

**3.实现深拷贝的方法：**

**a) 递归**

// 递归实现深拷贝

const deepCopy = (obj) => {

    let deepObj = Array.isArray(obj) ? [] : {}

    for (let key in obj) {

        if (obj[key] && typeof (obj[key]) === 'object') {

            deepObj[key] = deepCopy(obj[key])

        } else {

            deepObj[key] = obj[key]

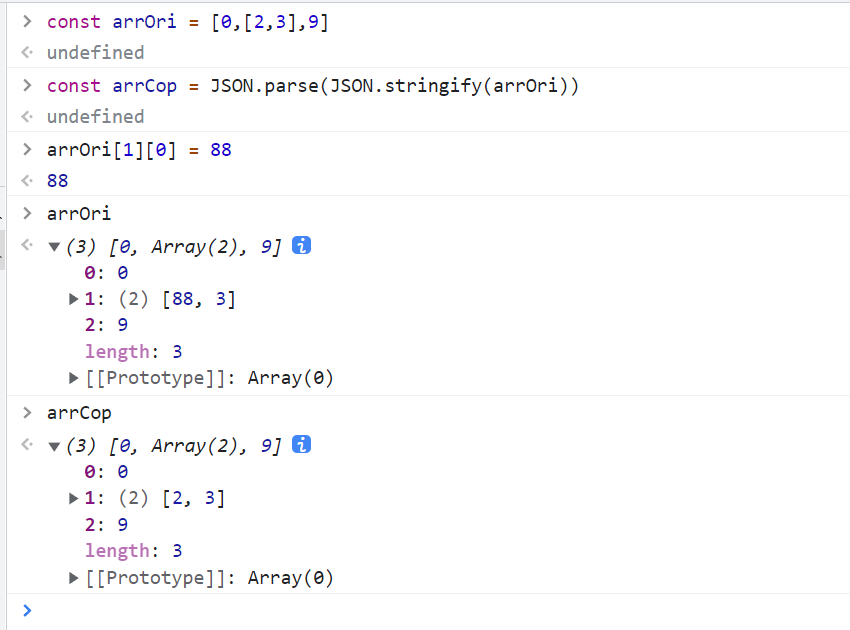
        }

    }

    return deepObj

}

**b) JSON. parse (JSON. stringify (obj))**

****

1. **一些相关面试题：（除涉及到拷贝的内存原理，还涉及构造函数等）([JS对象，原型链，构造函数面试题\_aiyx16521的博客-CSDN博客\_js原型链面试题](https://blog.csdn.net/cjmxcc/article/details/109290599))**

**//**

**var a = [1, 3]**

**var b = a**

**a = [20,1]**

**console.log(b) // [1, 3]**

**//**

**function change(arr) {**

**for(var i=0;i<arr.length;i++){**

**arr[i]+=2**

**}**

**}**

**var arr；**

**arr = [2,6]**

**change(arr)**

**console.log(arr) // [4, 8]**