**数据类型分类：**

1.基本类型值（原始值）：number string boolen null undefined Symbol 存储在**栈**内存中

2.引用类型值（对象，函数）Object Array Regexp Date……. Function 开辟**堆**内存存储

浏览器加载页面会形成一个ECStack（执行环境栈）供代码执行，产生一个EC(GLOBAL)全局执行上下文，将整个EC(GLOBAL)进ECStack执行。

**type of null 是”Object”的原因:** 不同对象底层是用二进制表示的，js将前三位都是0判定为Object类型，null的存储二进制是000。

**000 对象类型**

**1 整型号**

**100 字符串**

**110 布尔类型**

**==：**

对象==对象 比较地址

对象==字符串 对象变为字符串

NaN==NaN NaN和自己以及其他值都不相等

null==undefined 相等，与其他值不相等

剩下情况只要两边类型不同，将两边都转换为数字再比较

**1.如果有一个操作数是布尔值，则在比较相等性之前先将其转换为数值——false转换为0，而true转换为1；**

**2.如果一个操作数是字符串，另一个操作数是数值，在比较相等性之前先将字符串转换为数值**

**3.** **如果一个操作数是对象，另一个操作数不是，则调用对象的valueOf()方法，用得到的基本类型值按照前面的规则进行比较，如果对象没有valueOf()方法，则调用 toString()**

4.如果两个操作数都是对象，则比较它们是不是同一个对象，如果两个操作数都指向同一个对象，则相等操作符返回 true；否则， 返回false

**垃圾回收机制**

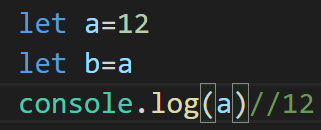
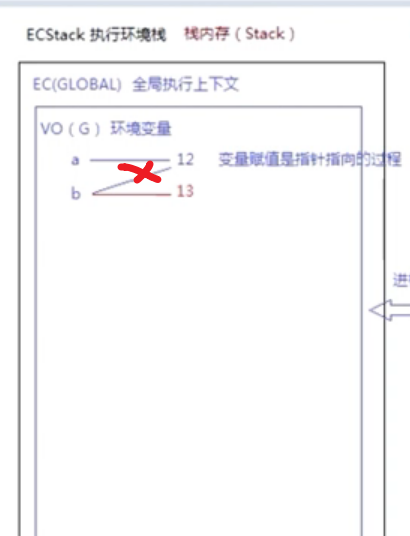
1.标记清除法 谷歌（周期性自动回收）

2.计数引用法 IE 回收计数为0的。弊端：计数不准确，导致有些内存未被清除（**内存泄漏**）

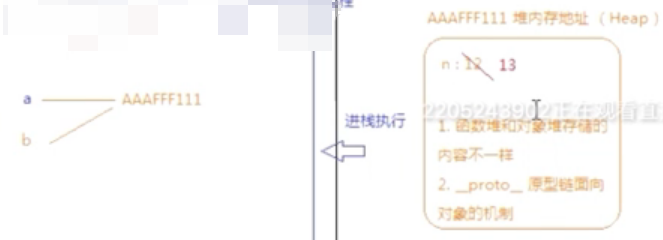
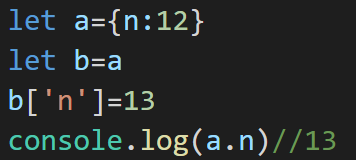
**堆**内存释放 变量=null

**栈**内存释放 函数执行完，形成的执行上下文，没有东西被上下文以外的内容占用，此上下文就会从执行环境栈中移除（释放），如果有被占用，则压缩到栈的底部（没有被释放=>闭包）

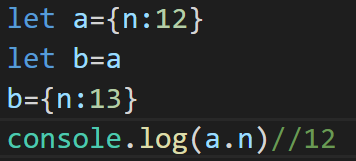
1.

2.

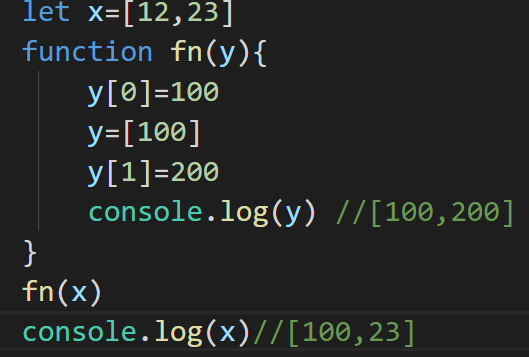


3.





4.



5.

