1.break: 终止 中断

在循环中 表示终止当前循环,循环余下次数不再执行

2**continue**;*//终止本次循环 跳入下一次循环*

Continue和break之后不能写任何代码 之后表示处在同一个顺序结构后面

While循环

格式:

初始化变量  
while(判断条件){  
 循环体语句  
  
 更新循环变量  
  
}

Do while循环 ---了解

无论第一次的判断条件是true还是false 会上来首先执行一遍 然后再去判断

初始化变量  
do{  
 循环语句体  
 更新循环变量  
  
}while(循环判断条件);

数组

概念:

字面意思: 一组数

为什么要学习数组:

之前我们存储数据用变量,如果存储比较少的数据还可以,如果说数据量比较大

比如 我们班所有同学的 身高 年龄 座位号 等等等等…. 用我们现在所学的知识,需要声明好多变量, 不方便查询 修改

意义: 为了更方便的管理多个数据

特点:

* 数组：数据的有序列表，可以存放任意类型的数据，数组的大小可以动态调整。

//创建一个空数组 长度为0  
var arr=[];

//创建一个有初始值的数组 长度是系统默认计算  
var arr1=[1,2,3];

*创建一个空数组*

var arr3=new Array();  
 console.log(arr3);

//创建一个具有指定长度的数组

*我们没有给数组的元素赋值 默认undefined*

var arr4=new Array(5);

//打印数组长度  
console.log(arr.length);

//向数组插入值 操作数组 使用角标(索引) 索引值从0开始  
//如果创建数组时没有指明数组长度 那么数组长度是根据放入的元素数量相等  
  
arr[0]=5;  
arr[1]='xx';  
arr[2]=true;  
arr[3]=6.32;  
console.log(arr);  
//修改指定位置的元素值  
arr[0]=98;  
console.log(arr);  
//获取指定位置的元素  
console.log(arr[1]);

*数组的长度等于最后一个元素的索引值+1*

如果new Array()括号里面直接写一个整数 就是指明数组长度 其他值都是给数组插入元素

var *arr4*=new Array(5);

数组遍历

遍历: 把所有元素挨个取出来

循环改进:  
 //数组最后一位的索引值 是数组长度-1  
 //第一种遍历 利用for循环的i递变 取出每一个元素  
 for(var i=0;i<arr.length;i++){  
 var str=arr[i];  
 console.log(str+" kkk");  
  
 }

*//for in 循环 专门遍历数组的  
//第二种遍历方式 for in 遍历数组 把数组的索引值放入我们自己声明的变量里面*for(var *index* in *arr*){  
 console.log(*index*);  
 console.log(*arr*[*index*]);  
 console.log("-------------------")  
  
}