复习

匿名函数funchtion (){}

创建函数 var fn=function(){} fn()

函数表达式 先创建，再调用

函数声明 存在函数提升

匿名自调用 (function(){ })(1,2);

匿名函数以参数传递 (回调函数) add(function(){return 1;})

对象

属性和方法

原生对象，宿主对象，自定义对象

对象字面量

var person={name:’tom’,age:18, say:function(){ }}

内置构造函数

var person=new Object();

new 所调用的函数返回一个对象

person.name=’tom’

对象中成员的访问

person.age/persong[‘age’]/persong.say()

for(var key in person){ person[key] }

0.检测对象中是否含有某个属性

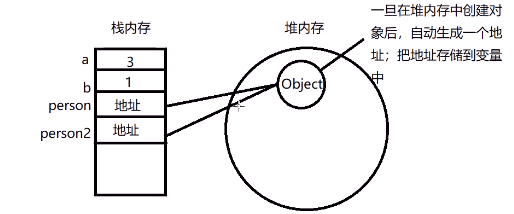
‘属性名’in 对象 存在->true 不存在->false

对象.hasOwnProperty(‘属性名’) 存在->true 不存在->false

对象.属性名===undefined 存在->false 不存在->true

‘属性名’ in 对象 true -> 存在 false -> 不存在

1.对比原始类型的引用类型存储



原始类型存储在栈内存中

引用类型存储在堆内存中，自动生成一个地址，保存在了栈内存。

如果要销毁内存中的数据 ，只需要设置为null即可

var s1=’tom’;

var s2=’jerry’;

.....

2.数组

数组是有多个元素(数据)组成的集合。

每个元素就代表一个数据

1. 创建数组----数组字面量

[元素1,元素2...]

访问数组中的元素，使用下标，下标是从0开始。

数组[下标]

（3）内置构造函数

new Array(元素1,元素2...);

new Array(5); 5代表初始化的长度，可加可不加，后期可以添加更多个元素。

练习：创建数组保存若干个课程名称

创建数组：初始化长度为5.添加篮球场上的五个位置

数组的长度

数组名称.length 获取数组中元素的个数

作用：在数组的末尾添加一个新的元素

数组名称[数组名称.length]=值;

数组中最后一个元素的下标

数组.length-1

练习：创建一个空数组。使用数组长度添加若个国家名称

练习：创建数组，在末尾添加4个国家名称，打印最后一个元素

var country=new Array('中国');

country[country.length]='小日本';

country[country.length]='泰国';

country[country.length]='美国';

country[country.length]='韩国';

//console.log(country);

console.log(country[country.length-1]);

1. 数组的分类

索引数组：以0以上的整数作为下标 //var cheji=[28,87,98];

关联数组：以字符串作为下标，只能单独的添加元素

//var person=[];

person['name']='tom';

|  |
| --- |
| var emp=[];  emp[‘ename’]=’tom’; |

1. 遍历数组元素

遍历数组的元素使用遍历下标

for-in

|  |
| --- |
| for(var key in 数组){  key要遍历的元素的下标  数组[key] 下标的对应的元素  console.log(key,数组[key]);  }  既可以遍历索引数组，也可以遍历关联数组 |

练习：创建数组，包含有多个学生的成绩，获取成绩

for循环

|  |
| --- |
| for(var i=0;i<数组.length;i++){  i 代表下标 0~数组长度-1；  数组[i] 代表下标对应的元素  }  只能遍历索引数组。 |

练习：创建数组。包含有多个姓名，把tom全部改成‘汤姆’；

创建数组

遍历得到每一个元素

判断每个元素是否为tom，如果是。重新赋值为‘汤姆’在遍历后，打印数组

练习：创建数组。包含多个姓名。查询tom出现的次数；

创建数组

遍历数组

初始化变量的次数为0

判断每个元素是否为tom。如果是次数加1，

打印遍历

创建数组。包含多个 数字。获取这组数字的最大值

声明变量为用于保存最大值。默认把第一个元素放进去

遍历数组中的元素 13 85 74 46 95 100

创建函数 getAvg，传递一个参数(一组工资)，在返回工资的平均值

实参就是一个数组 。包含多个工资；

函数体中，计算 元素的和。求平均

平均=总和/元素长度

练习：创建数组，保存若干个成绩，使用遍历数组元素，让每个成绩加10分

var score=[74,80,91,73,82];

//遍历数组中的元素，得到每一个成绩

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

//console.log(score[i]);

//让每一个成绩score[i] 加10，

//score[i]+=10;

score[i]=score[i]+10

}

console.log(score);

练习：创建数组，保存若干个汽车品牌，把品牌名称‘宝马’替换为‘BMW’

var car=['大众','宝马','奔驰','兰博基尼','宝马'];

//遍历数组元素，得到每个品牌名称

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

//判断元素是否为宝马

if(car[i]=='宝马'){

//替换为BMW

car[i]='BMW';

}

}

console.log(car);

练习：创建函数getTotal,传递1个参数(数组)，返回工资的总数。

getTotal([5000,8000,6000])

/\*function getTotal(salary){

//遍历数组salary中的元素

for(var i=0,sum=0;i<salary.length;i++){

sum+=salary[i];

}

return sum;

}

console.log(getTotal([5000,6500,8400,9100]));

\*/

练习：创建函数getCount,传递2个参数(数组，字符串)，返回字符串在数组中出现的次数。

getCount([‘tom’,’jerry’,’kate’,’tom’],’tom’) -> 2

function getCount(arr,str){

//遍历数组元素，获取每个元素

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

//如果元素是str,让次数加1

if(arr[i]==str){

count++;

}

}

return count;

}

console.log(getCount(['tom','jerry','tom','kate','king','tom'],'jerry'));

练习：创建函数getMax,传递1个参数(数组)，包含有若干个数组，返回最大值。

getMax([70,68,92,43,85]);

function getMax(score){

//初始化一个变量用于存储最大值

//假设第1个元素就是最大值

var max=score[0];

//遍历第2个元素到最后

for(var i=1;i<score.length;i++){

//用max和每个元素进行比较

//如果max小于任意元素，则把该元素放到max中

if(max<score[i]){

max=score[i];

}

}

return max;

}

console.log(getMax([70,80,90,100]));

4.数组中的方法(API)

API 应用程序编程接口，预定义好的方法/函数

toString() 将数组中的元素转为字符串，按逗号分隔。

join(‘-’) 将数组中的元素转为字符串，指定分隔符,默认是逗号

concat(arr1,arr2...) 拼接两个或者更多个数组。

slice(start,end)截取数组中的元素，start开始的下标，end结尾的下标，不包含end本身，如果负数表示倒数。返回 一个数组；

练习：创建数组a~g，每个字母是一个元素，分别截取cd，，f d，把截取的结果拼接成一个新数组

splice(start,count,value1,value2...) 在删除数组中的元素，start开始的下标，count删除的数量，value1...在删除位置插入的元素。返回删除的元素。原数组会发生变化

练习：创建数组a~h，每个字母是一个元素，删除d e。替换f为m，在下标为1的位置插入字母z

reverse() 翻转数组中的元素

sort() 对数组中的元素进行排序，默认是按照Unicode 码从小到大

|  |
| --- |
| 对数字排序  sort(function(a,b){  return a-b; //从小到大  //return b-a; //从大到小  }) |

复习今天内容，

使用遍历数组翻转数组 中的元素

['a','b','c','d'] -> [ 'd', 'c', 'b', 'a' ]

使用冒泡排序数组中的数字从小到大排序 循环嵌套

练习：创建3个数组，每个元素含有一个字母，拼接这3个数组，再将数组转为字符串。

var arr1=['a','b','c'];

var arr2=['d','e'];

var arr3=['f','g'];

var newArr=arr1.concat(arr2,arr3);

console.log(newArr.join('-'));

练习：创建数组，每个元素有1个字母组成a~h,分别截取bc和fg,拼接成一个新数组。

var arr=['a','b','c','d','e','f','g','h'];

var arr1=arr.slice(1,3);

var arr2=arr.slice(-3,-1);

console.log(arr1.concat(arr2));

练习：创建数组，保存a~h，每个字母为一个元素，删除c,d;替换f为m,在下标为1的位置插入元素z

var arr=['a','b','c','d','e','f','g','h'];

arr.splice(2,2);

arr.splice(3,1,'m');

arr.splice(1,0,'z');

//console.log(arr);