一.对象

1.检测属性是否存在

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

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

'属性名' in 对象 true->存在 false->不存在

练习：创建商品对象，包含的属性有编号，标题，价格；如果产地属性不存在，则添加该属性；如果价格属性存在，则打八折。

2.方法

本质上就是一个函数

|  |
| --- |
| var person={  play: function(){  this //指向调用方法的对象  }  }  person.play(); //调用方法 |

练习：创建一个圆对象，包含的属性有半径和圆周率，添加两个方法，分别计算圆的周长和面积，并返回结果。

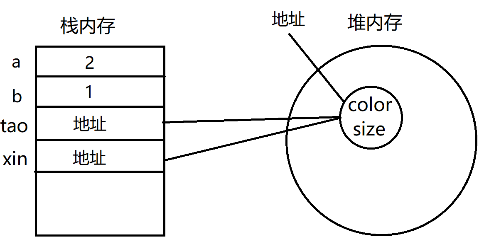
二.数据的存储

1.原始类型

存储在栈内存中

2.引用类型

存储在堆内存中，会自动生成一个地址，把地址保存到栈内存中



3.数据的销毁

原始类型的数据会自动销毁，引用类型的数据不会自动销毁，只有不被任何地址所指向才会自动销毁

null表示一个空地址，不指向任何堆内存数据

如果要销毁引用类型数据，只需要赋值为null即可

三.数组

是一组数据的集合，每个数据称为元素

1.数组字面量

[元素1, 元素2, ...]

练习：创建数组，保存一组工资；创建数组，保存一组商品名称

2.访问元素

下标(索引)：数组自动为每个元素添加的编号，是一个从0开始的整数

数组[下标]

如果访问下标不存在的元素值为undefined

练习：创建数组，包含若干个城市，修改其中的元素，在最后添加新的元素

3.数组的长度属性

数组.length 获取数组元素的个数

(1)在数组末尾添加元素

数组[数组.length] = 值

练习：创建空数组，使用数组的长度往数组中添加若干个汽车品牌名称

(2)清空数组

数组.length=0

数组就变成了空数组 [ ]

4.构造函数创建数组

new Array(元素1, 元素2,...)

new Array(n) //创建数组，初始化长度为n，n是>=0整数

练习：创建数组，包含有一组国家的名称；创建数组，初始化长度为5，添加篮球场上5个位置

大前锋 小前锋 中锋 得分后卫 控球后卫

5.数组的分类

索引数组：以>=0的整数作为下标

关联数组：以字符串作为下标，需要单独的添加元素，无法获取数组长度

|  |
| --- |
| 对比数组和对象的区别  相同点：两者都可以存储一组数据  不同点：对象存储的每个数据是在属性中，数组是在元素中；对象中的一组数据是无序的，数组中的数据是有序的，可以进行排序。 |

6.遍历数组

依次访问数组中的每个元素

|  |
| --- |
| for-in循环  for(var k in 数组){  k 下标  数组[k] 下标对应的元素  } |
| 循环(推荐方法)  for(var i=0;i<数组.length;i++){  i 下标  数组[i] 下标对应的元素  } |

练习：创建数组，包含一组成绩，遍历数组，计算出总成绩和平均成绩

7.API

JS中预定义的函数和方法

toString() 将数组转为字符串

join() 将数组转为字符串，可以指定元素之间分隔的符号

课后任务

(1)复习今天内容，整理思维导图

(2)练习：遍历数组，翻转数组元素

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

练习：遍历数组，将一组数字进行从小到大的排列(选做)

[23,9,78,6,45] [6,9,23,45,78]

(3)预习js第8天

http://www.codece.com/archives/107