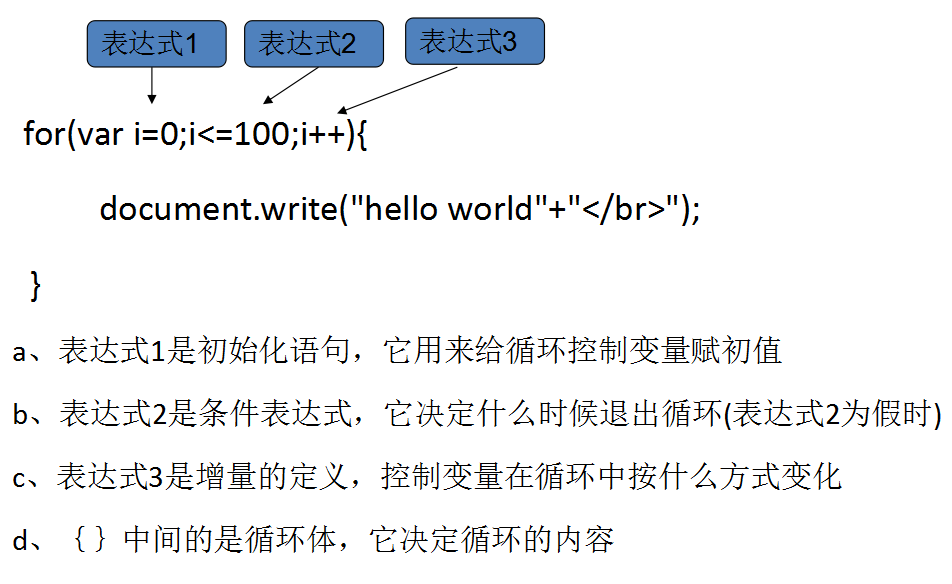
循环语句：for循环

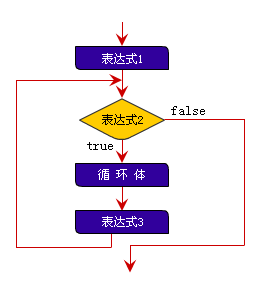
for（表达式1;表达式2;表达式3）

{

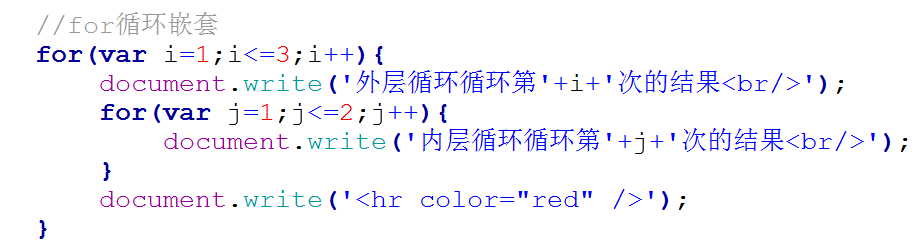
循环体

}





for循环嵌套

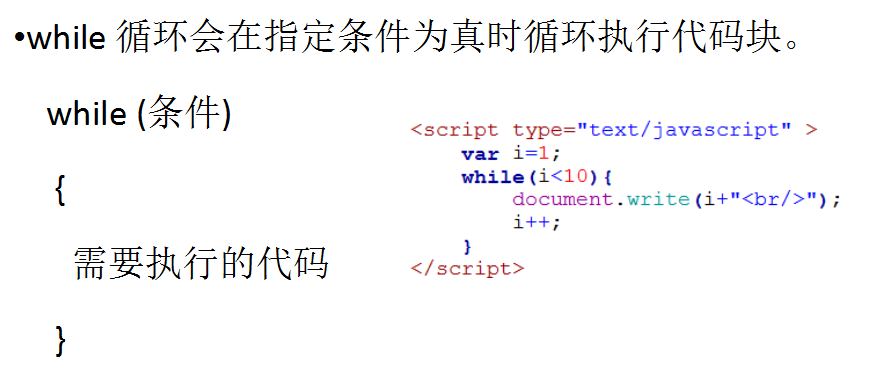


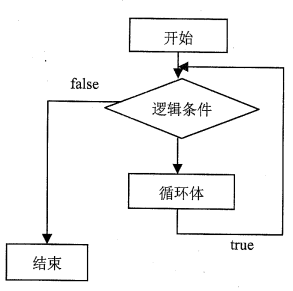
练习1——打印100以内的奇数和

练习2——99乘法表

练习3——打印等腰三角形

循环语句：while循环





练习4——打印菱形

练习5——打印100以内的质数，并求和

练习6——公鸡5元一只，母鸡3元一只，小鸡1元2只；100元买100只鸡，求有多少种买法？并求出每种买法各种鸡的数量

循环语句：do while循环

do/while 循环是 while 循环的变体，该循环会在检查条件是否为真之前执行一次代码块，然后如果条件为真的话，就会重复这个循环

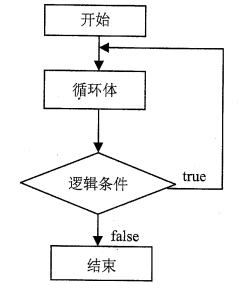
do

{

需要执行的代码

}

while (条件)



For，while do while三个循环的区别

For：先判断，在执行，并且知道循环次数

While：先判断，在执行，不知道循环次数

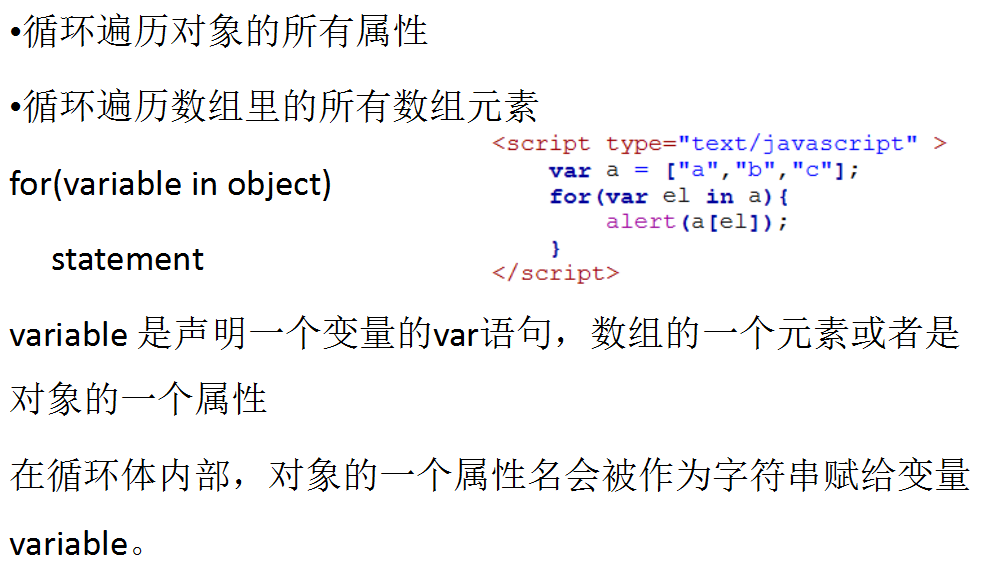
Do while：先执行一次，在判断，也是不知道循环次数

循环中断:

break//彻底结束循环，跳出当前，不在继续执行

continue//结束本次循环进行下一次循环

循环语句：for in



为什么要用Array对象:

一个变量存储一个值（字符串），如果用变量存储多个字符串

var str1 = "HTML";

var str2 = "CSS";

var str3 = "JavaScript";

var str4 = "jQuery";

我们可以使用“数组”来存储一组“相同数据类型”的数据结构

var arr = new Array("HTML","CSS","JavaScript","jQuery",12);

也可以是不同数据类型

Array对象：数组是一系列的变量，与其他强类型语言不同的是JavaScript中数组元素的类型可以不相同。

数组对象的作用是：使用单独的变量名来存储一系列的值。

数组使用下标方式来获取某一项数值

数组创建：

方法1：新建一个长度为0的数组

var 数组名 = new Array();

方法2：创建空数组

var 数组名=[];

方法3：新建指定长度的数组并赋值

var 数组名=new Array(元素1,元素2,...元素n);

创建Array对象的方法区别如下：

方法1和方法2的元素的个数是不确定的，用户可以在赋值时任意定义；

方法3的数组长度由数组元素的个数决定。

数组的常用属性和方法：

属性：

length ：设置或返回数组中元素的数目

方法：

join( )：把数组的所有元素放入一个字符串，通过一个的分隔符进行分隔

sort( )：对数组的元素进行排序，reverse：将数组反序

push( )：末尾追加元素,并返回新的长度

splice(start,deleteCount,val1,val2,...)：从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...

pop( )：删除原数组最后一项，并返回删除元素的值；如果数组为空则返回undefined

shift( )：删除原数组第一项，并返回删除元素的值；如果数组为空则返回undefined

unshift( )：将参数添加到原数组开头，并返回数组的长度

slice(start,end)：返回从原数组中指定开始下标到结束下标之间的项组成的新数组

数组的应用：

选择排序：是一种简单直观的排序算法。它的工作原理是每一次从待排序的数据元素中选出最小（或最大）的一个元素，存放在序列的起始位置，直到全部待排序的数据元素排完。

冒泡排序：是一种计算机科学领域的较简单的排序算法。它重复地比较要排序数列的大小，一次比较两个元素，如果他们的顺序错误就把他们交换过来。比较数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。

练习7——请给数组（45,23,65,21,12,78,55）按从小到大的顺序排序

练习8——var data=new Array(1,2,3);

程序要求打印出数组每个数组元素相互之间各种可能性加法(包括元素自已本身加法)

1+1=2,1+2=3,1+3=4,

2+2=4,2+3=5,

3+3=6

打印出以下结果

Array

(

[0] => Array

(

[0] => 2

[1] => 3

[2] => 4

)

[1] => Array

(

[1] => 4

[2] => 5

)

[2] => Array

(

[2] => 6

)

)