

# 1. 数组的概念

数组是指一组数据的集合，其中的每个数据被称作元素，在数组中可以存放任意类型的元素，数组是一种将一组元素存储在单个变量名下的优雅方式

## 2. 创建数组

### 2.1. 数组的创建方式

js中创建数组有两种方式：

- 利用new创建数组
- 利用数组字面量创建数组

### 2.2. 利用new创建数组

```
var 数组名=new Array();  
var arr=new Array();
```

### 2.3. 利用数组字面量来创建数组

```
var 数组名=[];
```

- 数组里面的数据称为数组元素

### 2.4. 数组元素的类型

数组中可以存放任意类型的数据，例如：字符串，数字，布尔值等

## 3. 获取数组元素

### 3.1. 数组的索引

索引（下标）：用来访问数组元素的序号（数组下标从0开始）

数组可以通过索引来访问、设置、修改对应的数组元素，我们可以通过“数组名[索引]”的形式来获取数组中的元素。

这里的访问就是获取得到的意思

## 4. 遍历数组

怎么取出数组中的每一项

循环

遍历：就是把数组中的每个元素从头到尾都访问一次（类似我们每天早上学生的点名）

### 4.1. 数组的长度

使用“数组名.length”可以访问数组元素的数量（数组长度）

## 5. 数组中新增元素

可以通过修改length长度以及索引号增加数组长度

### 5.1. 通过修改length长度来新增数组长度

可以通过修改length长度来实现数组扩容的目的

length属性是可读写的

```
var arr=['red','blue','green','yellow'];
arr.length=7;
```

其中索引号4,5,6号的空间没有给值，就是声明变量为给值，默认值就是undefined

### 5.2. 通过修改数组索引新增数组元素

可以通过修改数组索引的方式追加数组元素

不能直接给数组名赋值，否则会覆盖掉以前的数据

这种方式也是我们最常用的一种方式

## 冒泡排序

是一种算法，把一系列的数据按照一定的顺序进行排列显示(从小到大或从大到小)

### (1)一共需要的趟数 我们用外层for循环

5个数据我们一共需要走4趟

长度就是数组长度减去1 `arr.length-1`

### (2)每一趟交换次数我们用里层for循环

第一组交换4次

第二组交换3次

第三组交换2次

第四组交换1次

长度就是数组长度减去次数

但是我们次数是从0次开始的 所以最终`arr.length-i-1`

### (3)交换2个变量就好了

## 6. 函数的概念

在js里，可能会定义非常多的相同代码或者功能相似的代码，这些代码可能需要大量重复使用

虽然for循环语句也能实现以些简单的重复操作，但是比较具有局限性，此时我们讲究可以使用js中的函数

函数：就是封装了一段可以被重复调用执行的代码块，通过此代码块可以实现大量代码的重复使用

## 7. 函数的使用

函数在使用时分为两步：声明函数和调用函数

### 7.1. 函数的声明

```
function 函数名(){  
    //函数体;  
}
```

- function 声明函数的关键字，全部小写
- 函数是做某件事情，函数名一般是动词
- 函数不调用自己不执行

## 7.2. 调用函数

//调用函数

函数名(); //通过调用函数名来执行函数体代码

调用的时候不要忘记加小括号

注意：声明函数本身并不会执行代码，只有调用的函数的时候才会执行函数体代码

## 7.3. 函数的封装

函数的封装就是把一个或者多个功能通过函数的形式封装起来，对外只提供一个简单的函数接口  
简单理解：就是类似于电脑配件整合组装到机箱中

# 8. 函数的参数

## 8.1. 形参和实参

在声明函数时，可以在函数名称后面的小括号中添加一些参数，这些参数被称为形参，而在调用该函数时，同样也需要传递相应的参数，这些参数被称为实参

形参：形式上的参数，函数定义的时候传递的参数，当时并不知道是什么

实参：实际上的参数，函数调用的时候传递的参数，实参是传递给形参的

参数的作用：在函数内部某些值不能固定，我们可以通过参数在调用函数时传递不同的值进去

## 8.2. 形参和实参个数不匹配的问题

实参的个数等于形参的个数：输出正确结果

实参个数多于形参个数：只取到形参的个数

实参的个数少于形参的个数：多的形参定义为undefined结果为NaN

注意：在JavaScript中，形参的默认值是undefined

## 9. 函数的返回值

### 9.1. return语句

有的时候，我们会希望函数将值返回给调用者，此时通过return语句就可以实现

```
function getResult() {  
    return 666;  
}  
getResult(); //getResult()=666  
console.log(getResult());
```

### 9.2. return终止函数

return 语句之后的代码不会被执行

```
function getSum(num1,num2){  
    return num1+num2;           //return后面的代码不会被执行  
    alert('我是不会被执行的');  
}  
console.log(getSum(1,2)); //3
```

return只能返回一个值，如果用逗号隔开多个值，以最后一个为准

### 9.3. 函数没有return返回undefined

函数都是有返回值的

如果有return则返回return后面的值

如果没有return则返回undefined

### 9.4. break,continue,return 的区别

break：结束当前的循环体（如：for、while）

continue：跳出本次循环，继续执行下次循环（如：for、while）

return: 不仅可以退出循环, 还能够返回return语句中的值, 同时还可以结束当前的函数体内的代码