

Day05_数组和对象

讲师: 聂剑峰
联系方式: 610567895

目 录

- 数组Array
- 数组的定义
- 数组的使用
- 数组的常用函数
- 练习
- 排序算法-冒泡排序(扩展)
- 对象Object
- 作业

➤ 数组的概念

1. JS数组的概念

数组的字面意思就是一组数据，一组数据（一般情况下是相同类型的数据，不一定是数字）

2. 数组的作用

使用单独的变量名来存储一系列的值，数组是特殊的变量，它可以同时保存一个以上的值。

如上节课中我们学过的`arguments`就是一个数组，里面保存了多个参数。

3. 为什么要使用数组？

如果您有一个项目列表（例如汽车品牌列表），在单个变量中存储这些品牌名称是这样的：`var cars1="Volvo"; var ="BMW"; var = "Benz";`

不过，假如您希望对变量进行遍历并找出特定的那个值？或者如果您需要存储300个汽车品牌，而不是3个呢？

解决方法就是创建数组！

➤ 数组的定义

1, 数组的定义

方式一:

`new Array (参数, 参数,...)` : 只有一个数字参数时是数组的长度
(`new`可以省略,但一般尽量写上)

如:

`var arr = new Array();` //定义一个空数组

`var arr = new Array(10);` //创建一个包含 10 个元素的数组, 没有赋值

`var arr = new Array("芙蓉姐姐",30);` //创建一个数组有两个元素

方式二:

`var arr = [1,2,3,4,5];` //字面量定义方式

2, 数组元素的获取(访问)

`arr[0]`: 表示数组的第一个元素, 0是下标, 也叫索引

`arr[1]`: 表示数组的第二个元素, 1是下标

....

➤ 数组的定义

3, 数组的长度

数组长度(数组元素的个数): **arr.length**,length属性,不止是只读的,也可以设置如:

```
var colors = new Array("red", "blue", "green");  
colors.length=2;  
console.log(colors[2]);
```

数组的下标从0开始,所以数组的最后一个元素的下标为: **arr.length - 1**

注意: 下标可以是变量或表达式

➤ 数组的赋值

数组的赋值

给数组赋值，就是给数组的元素赋值，需要通过给数组的每个元素一一赋值，

如：`arr[0] = 20;` //让数组的第一个元素的值为20;

`arr[1] = 12;` //让数组的第二个元素的值为12;

以下为通过for循环给数组的每个元素赋值，赋成下标的平方。

```
for(var i=0;i<10;i++){  
    arr[i] = i*i  
}
```

以下为通过for循环给数组的每个元素赋值，赋值一个随机数:Math.random()

```
for(var i=0;i<10;i++){  
    arr[i] = Math.random();  
}
```

Math.random()的范围是:0~1(包括0, 不包括1)

➤ 数组的使用

数组的使用

使用数组就是在使用数组的每个元素，因为数组相当于若干个相同类型的变量。

遍历数组:

之前我们讲过通过下标获取单个数组元素,但有时候我们需要批量使用数组,这个时候我们需要遍历整个数组.

1, 普通for循环

```
for(var i=0; i<5; i++){  
    console.log(arr[i]);  
}
```

2, for...in遍历: 用于遍历数组或者对象

```
for(var i in arr){  
    console.log(arr[i]);  
}
```

➤ 数组的使用

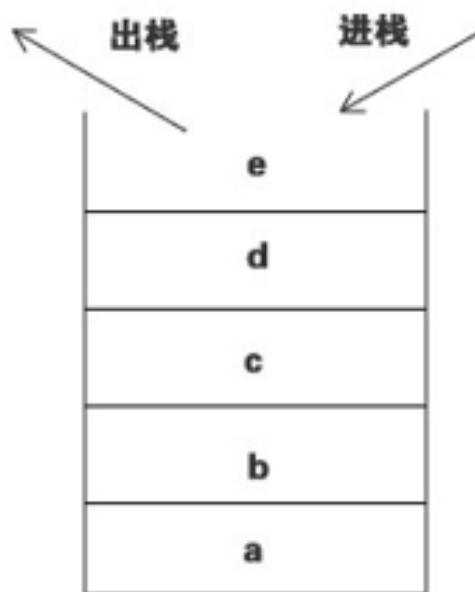
示例:

- 1,求数组中所有元素的和
- 2,让数组中的元素交换位置 (重要)
- 3,求数组的最大数和最小数 (重要)
- 4,求数组的最小数的下标 (重要)

➤ 数组的常用函数

1, 栈方法 : push() 和 pop()

ECMAScript 数组提供了一种让数组的行为类似于其他数据结构的方法。可以让数组像栈一样，可以限制插入和删除项的数据结构。栈是一种数据结构(后进先出)，也就是说最后添加的元素最早被移除。而栈中元素的插入(或叫推入)和移除(或叫弹出)，只发生在一个位置——栈的顶部。ECMAScript 为数组专门提供了 `push()` 和 `pop()` 方法。



➤ 数组的常用函数

1, 栈方法: push() 和 pop()

push()方法: 接收任意数量的参数, 把它们逐个添加到数组的末尾, 并返回修改后数组的长度;

pop()方法: 从数组末尾移除最后一个元素, 减少数组的 length 值, 然后返回移除的元素;

例如:

数组末尾添加一个元素, 并且返回长度

```
console.log(arr.push('张家界'));
```

移除数组末尾元素, 并返回移除的元素

```
var b = arr.pop();
```

➤ 数组的常用函数

2, 队列方法：push(), shift() 和 unshift()

队列在数组的末端添加元素，从数组的前端移除元素

push(): 和栈方法中的push是同一个，

shift(): 方法从数组前端移除一个元素，

unshift(): 方法从数组前端添加一个或多个元素。

例如:

数组末尾添加一个元素，并且返回长度

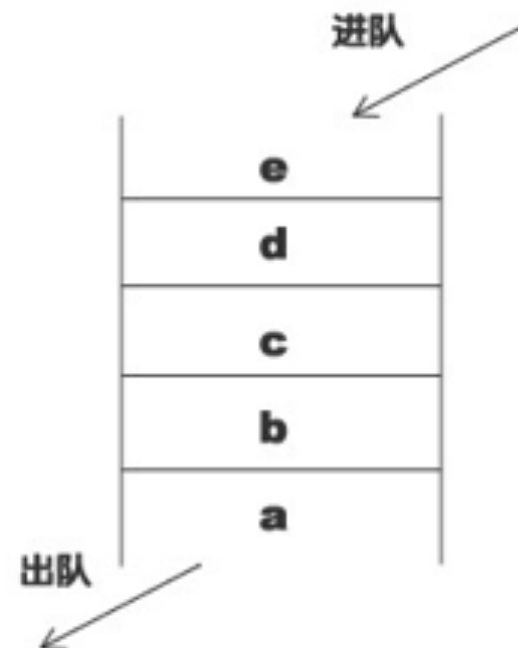
```
console.log(arr.push('深圳'));
```

移除数组开头元素，并返回移除的元素

```
console.log(arr.shift());
```

数组开头添加两个元素

```
console.log(arr.unshift('广东','深圳'));
```



➤ 数组的常用函数

3, 排序方法: reverse()和 sort()

reverse() : 逆向排序, 原数组也被逆向排序了

例如:

```
var arr = [1,2,3,4,5];  
console.log(arr.reverse()); //逆向排序方法 , 返回排序后的数组  
console.log (arr);
```

sort() : 从小到大排序 , 原数组也被升序排序了

例如:

```
var arr= [4,1,7,3,9,2];  
console.log(arr.sort()); //从小到大排序 , 返回排序后的数组  
console.log(arr);
```

➤ 数组的常用函数

sort()详解:

sort 方法的默认排序在数字排序上有些问题，因为数字排序和数字字符串排序的算法是一样的。我们必须修改这一特征，修改的方式，就是给 sort(参数)方法传递一个函数参数。

```
//如果一定要使用sort()进行排序,可以传递一个函数
function compare (value1, value2){
    if (value1 <= value2){
        return -1; //返回0和-1, 表示不交换值
    }
    return 1; //返回1, 表示会交换值
}
var box = [0, 1, 5, 10, 15];
var aseBox = box.sort(compare); //返回升序的数组
```

➤ 数组的常用函数

4, 数组操作的方法

ECMAScript 为操作已经包含在数组中的元素提供了很多方法。

concat(): 追加数据, 创建一个新数组, 不改变原数组

```
var arr = [2, 3, 4, '绿箭侠', '黑寡妇'];  
var arr2 = arr.concat('美队', '雷神');  
console.log(arr);  
console.log(arr2);
```

slice(): 不修改原数组, 将原数组中的指定区域数据提取出来

```
var arr = [2, 3, 4, "绿巨人"];  
var arr2 = arr.slice(1, 3); // 并没有修改原数组, 将原数组中的元素提取出来, 生成新数组, 取的是下标在区域: [1, 3)  
console.log(arr);  
console.log(arr2);
```

➤ 数组的常用函数

`splice()` : 截取原数组中指定的数据, 会改变原数组

```
var arr = [2, 3, 4, '绿箭侠', '黑寡妇'];
```

`var arr2 = arr.splice(2, 1);` //第一个参数代表我要开始截取的下标位置, 第二个参数截取的长度, 如果只有两个参数, 则表示删除操作

```
console.log(arr); // [2, 3, '绿箭侠', '黑寡妇'];
```

```
console.log(arr2); // 4
```

注意: `splice`和`slice`的区别

➤ 数组的常用函数

splice() 详解:

插入: 如果有3个或以上参数,且第二个参数(长度)为0,则表示插入

```
arr.splice(1, 0, "绿巨人", "冬兵");
```

//在下标为1的位置插入:"绿巨人","冬兵"

替换: 如果有3个或以上的参数, 且第二个参数(长度)不为0, 则表示替换

```
arr.splice(1, 1, "绿巨人", "冬兵");
```

//在下标为1的位置替换成:"绿巨人","冬兵"

删除: 如果只有两个参数, 则表示删除指定区域的数据

```
arr.splice(0, 2); //删除原数组的部分数据, 并返回截取的数据
```

join() : 连接数组中的元素, 并返回连接后的字符串

```
arr.join( "+" );
```


➤ 数组的常用函数

练习:

- 1, 不改变原数组, 取出数组[3,2,4,5,8,6,3,9]中的[5,8,6].
- 2, 在数组[1,2,3,4,6,7,8]中对应的位置插入5, 变成[1,2,3,4,5,6,7,8]
- 3, 将数组 [“我” , “是” , “一” , “只” , “笨” , “鸟”] 改成 [“我” , “是” , “一” , “只” , “聪” , “明” , “鸟”], 并打印出: “我是一只聪明鸟”
- 4, 删除数组[20,23,21,34,54,55,32]中的倒数第二个和倒数第三个元素

➤ 数组排序-冒泡排序(扩展)

用冒泡排序，对输入的6个数进行排序

思路：输入6个无序的数字，从头到尾依次比较相邻两个数字大小，若大数在前、小数在后，则交换两数位置，依次比较，使全部数据按从小到大排列

例如：将数组[8, 9, 7, 6, 5, 4]进行升序排序[4, 5, 6, 7, 8, 9]



➤ 数组排序-冒泡排序(扩展)

第二趟



第1次

第2次

第3次

第4次

第二趟排序后

第三趟



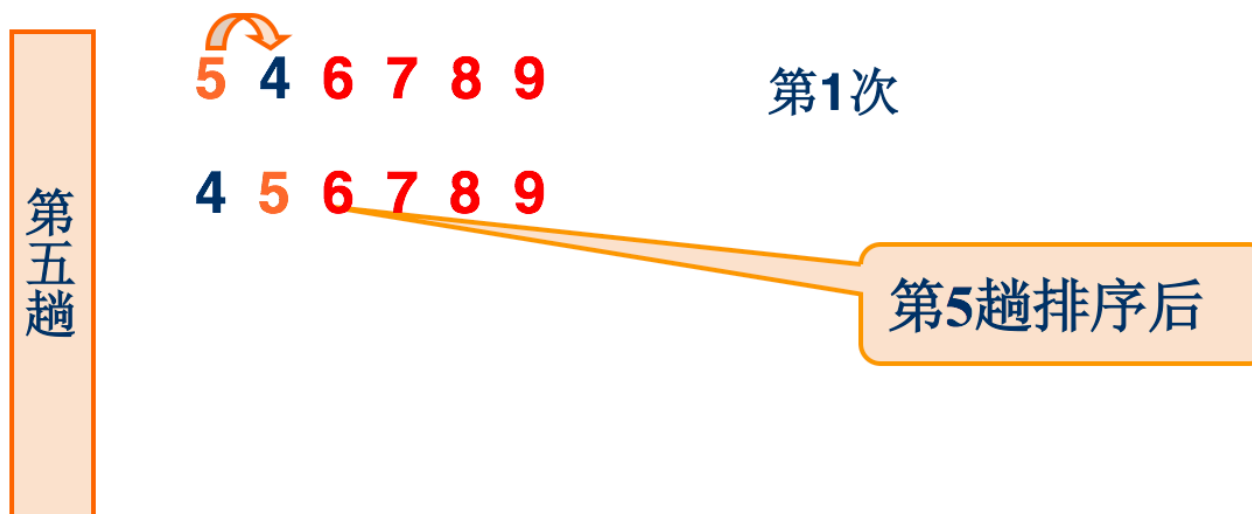
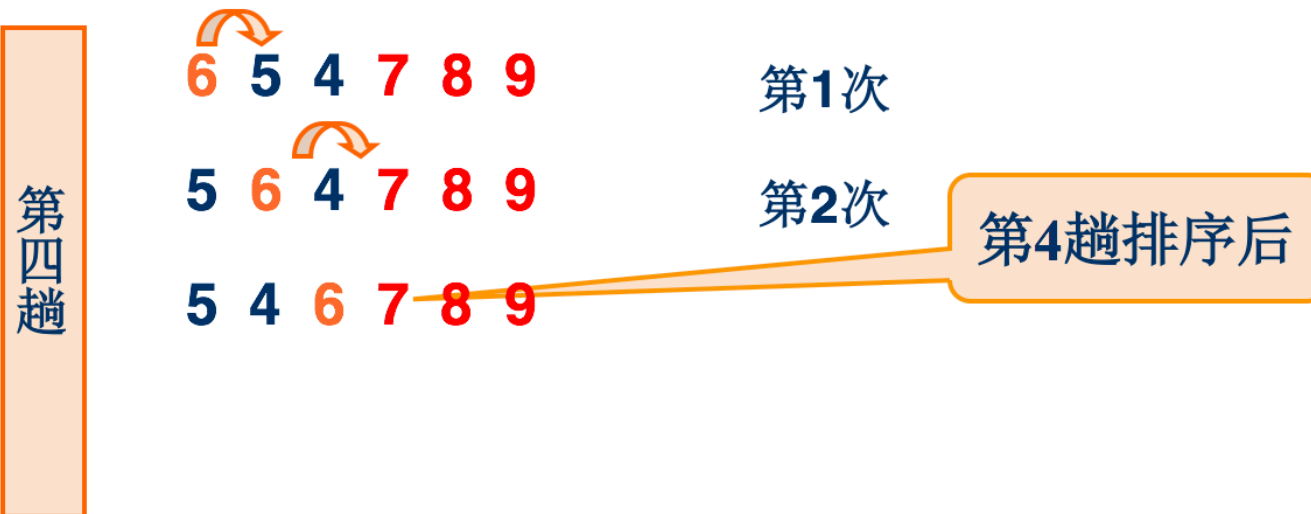
第1次

第2次

第3次

第三趟排序后

➤ 数组排序-冒泡排序(扩展)



➤ 数组排序-冒泡排序(扩展)

练习:

- 1, 将数组[1, 5, 6, 3, 2, 8, 9, 4] 降序排序
- 2, 请将数组[1,46,74,3,5,5]中的元素右移1位
- 3, 插数:在数组[1,46,74,3,5,5]的下标为2的位置插入一个数字8,
结果为[1,46,8,74,3,5,5]

➤ 对象Object

1. 对象的概念

对象Object是一种引用数据类型(在后期还会继续延伸对象的详细讲解)。
在 ECMAScript 中对象可以存储变量和函数(数据和功能)

2. 创建对象

方式一: 使用new

```
var obj = new Object();    //new方式  
obj.name = '张三';        //创建属性字段  
obj.age = 18;              //创建属性字段
```

new关键字可以省略

```
var obj = Object();        //省略了new关键字,不建议
```

方式二: 字面量方式

```
var obj = {  
    name: '张三', //创建属性字段,最后加逗号  
    age: 18  
};
```

➤ 对象Object

3. 属性字段也可以使用字符串形式

```
var box={  
    "name": "张三",    //也可以用字符串形式  
    "age": 28  
};
```

4. 使用字面量及传统赋值方式

```
var box={};    //字面量方式声明空的对象  
box.name='张三';    //点符号给属性赋值  
box.age= 18;
```

5. 两种属性输出方式

```
alert(box.age);    //点表示法输出  
alert(box["age"]);    //中括号表示法输出，注意引号
```

➤ 对象Object

6. 给对象创建方法

```
var obj={  
    run : function(){ //对象中添加方法(函数)run  
        retrun “正在跑步..”;  
    }  
}  
obj.run(); //调用对象中的方法
```

7. 使用 delete 删除对象属性 或 方法

```
delete obj.name; //删除属性  
delete obj.run; //删除方法
```


➤ 对象Object

练习:

- 1, 创建一个人的对象, 添加属性: 姓名 年龄 爱好 薪资期望, 并有一个打印自身信息的方法, 可以输出自身信息;
- 2, 创建一个锤子对象hammer, 有属性: 宽, 高, 重; 方法: 可以锤钉子
- 3, 有一辆50km/h车, 跑在一条1000km路上, 问多少小时跑完?
对象:

车Car 属性: 速度speed 50km/h

功能: 跑在路上runOnRoad(Road)

路Road 属性: 长度length 1000km

➤ 作业

- 1, 把课堂上的所有代码写一遍, 并掌握数组的定义, 赋值, 遍历, 常用函数等知识点.
- 2, 冒泡排序自己写一遍.
- 3, 给定一个含有n个元素的整型数组a, 求a中所有元素的和
- 4, 给定一个含有n个元素的整型数组a, 打印其中的最大值和最小值.
- 5, 给定一个不存在重复元素的整数数组, 例如[6,4,7,2,5,8]和一个数字, 例如10, 请设计一个函数找出两个元素(或同一个元素加自身), 并且使这两个数的和为给定数字, 并打印出来
例如[6,4,7,2,5,8]和数字10. 打印结果为: 6,4 2,8 5,5
- 6, 随机给出一个五位以内的数, 然后输出该数共有多少位, 并将每位的数字保存到数组中.
如:1342, 位数为:4, 数组为:[1,3,4,2]
- 7, 给定两个升序整型数组a和b, 打印其共同元素, 比如: a = [0, 1, 2, 3, 4], b = [1, 3, 5, 7, 9], 输出 1, 3
- 8, 有一个从小到大排好序的数组。现输入一个数, 要求按原来的规律将它插入数组中,
如: [2,3,4,56,67,98] //63, 100
- 9, 取出数组[1,3,1,4,2,3,6,2,6,1,5]中的重复项,存入一个新的数组,并从大到小排序

➤ 作业

10, 生成13位条形码(对之前的知识综合练习)

Ean-13码规则：第十三位数字是前十二位数字经过计算得到的校验码。

例如：690123456789 -> [6,9,0,1,2,3,4,5,6,7,8,9]

第十三位计算其校验码的过程为：

@前十二位的奇数位和 $6+0+2+4+6+8=26$

@前十二位的偶数位和 $9+1+3+5+7+9=34$

@将奇数和与偶数和的三倍相加 $26+34*3=128$

@取结果的个位数：128的个位数为8

@用10减去这个个位数 $10-8=2$

所以校验码为2（注：如果取结果的个位数为0，那么校验码不是（ $10-0=10$ ），而是0

实现函数ean13(n)计算验证码，输入12位条码，返回带验证码的条码。

例如：输入：692223361219输出：6922233612192

11, 开发一个标题为“FlipFlop”的游戏应用程序。它从1计数到100，遇到3的倍数就替换为单词“Flip”，5的倍数就替换为单词“Flop”，既为3的倍数又为5的倍数则替换为单词“FlipFlop”。

➤ 作业

对象作业:

找出下面题目中的对象,以及对象的属性和方法,并实现相应的功能

1. 有2个学生, 每个学生有3门课的成绩, (包括学生号, 姓名, 3门课成绩,语文,数学,英语), 计算出平均成绩.
2. Mountain: 从前有座山,山里有座庙;还有name,height(高度)
Temple: 庙里有个老和尚,有个小和尚,还有name,area(面积)
Monk: 和尚能挑水,讲故事,化缘;还有name,age,sex
3. 定义一圆Circle对象,包含半径r和圆心,圆心为“点” Point对象,point对象有x和y属性, 构造一圆, 该圆可以计算周长和面积;计算该圆的周长和面积?

THANKS