千锋教育
www.mobiletrain.org

千锋HTML5学院

第二阶段javascript课程课件

数组目录

为什么使用数组

当我们需要表示一组数据，或者叫做一次性定义很多相似的数字或变量时，就需要使用数组，如：表示一个班级学生的成绩，一年十二个月的销售数据等等。

数组的概念

□数组的概念

数组的字面意思就是一组数据，一组(一般情况下相同类型)的数据（不一定都是数字）

数组的作用是：使用单独的变量名来存储一系列的值。

数组的定义

□数组的定义

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）

```
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 arr2 = new Array("芙蓉姐姐",30); //创建一个数组有两个元素
var arr3 = [1,2,3,4,5];  //字面量定义方式
```

□使用数组元素（访问）；

arr3[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

……

数组的几个名词

◘ 数组的长度（length属性）：

数组元素的个数　arr.length

length属性,不是只读的，可以设置

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

◘ 数组的下标：

下标就是索引，即元素的序号，从0开始，下标最大取值是：数组的长度(length) -1；

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
}
//以下为通过循环给数组的每个元素赋值，随机数:Math.random()
for(var i=0;i<10;i++){
    arr[i] = Math.random();
}
```

数组的使用

- 不能一次使用整个数组，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：
  - 普通for循环

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

  - for...in 语句用于遍历数组或者对象的属性（快速遍历）

```
for(var i in arr){
        document.write(arr[i]);
}
```

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）

冒泡算法

- 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[9,8,7,6,5,4]进行升序排序

冒泡算法

冒泡算法


第四趟
6 5 4 7 8 9 第1次
5 6 4 7 8 9 第2次
5 4 6 7 8 9
第4趟排序后
第五趟
5 4 6 7 8 9 第1次
4 5 6 7 8 9
第5趟排序后

示例

- 将数组[1, 5, 6, 3, 2, 8, 9, 4] 降序排序.
- 请将数组[1,46,74,3,5,5]中的元素右移1位。
- 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] 。

扩展知识

常见算法（扩展知识）

- 选择法：(升序)通过比较首先选出最小的数放在第一个位置上，然后在其余的数中选出次小数放在第二个位置上,依此类推,直到所有的数成为有序序列。

如：已知原始数据: [47 33 61 82 72 11 25 47]

第一趟排序后: 11 [33 61 82 72 47 25 47]

第二趟排序后: 11 25 [61 82 72 47 33 47]

第三趟排序后: 11 25 33 [82 72 47 61 47]

第四趟排序后: 11 25 33 47 [72 82 61 47]

第五趟排序后: 11 25 33 47 47 [82 61 72]

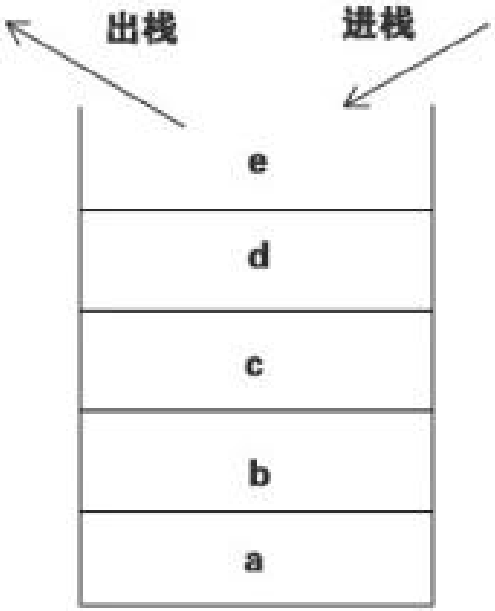
第六趟排序后: 11 25 33 47 47 61 [82 72]

第七趟排序后: 11 25 33 47 47 61 72 [82]

最后排序结果: 11 25 33 47 47 61 72 82

栈方法

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)，也就是说最新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。



栈方法

push()方法可以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()方法则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。

```
//数组末尾添加一个元素，并且返回长度
alert(box.push('盐城'));
//移除数组末尾元素,并返回移除的元素
box.pop()
```

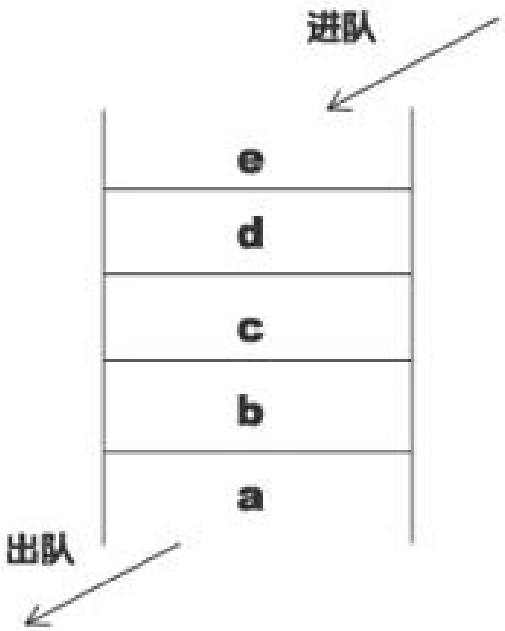
队列方法

队列在数组的末端添加元素，从数组的前端移除元素。通过

push() 向数组末端添加一个元素，

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。



alert(box.push('深圳'));	//数组末尾添加一个元素，并且返回长度
alert(box.shift());	//移除数组开头元素，并返回移除的元素
alert(box.unshift（‘广东’，‘深圳'));	//数组开头添加两个元素

重排序方法

数组中已经存在两个可以直接用来排序的方法：reverse()和sort()。

reverse()　　// 逆向排序, 原数组也被逆向排序了

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

sort()　　//从小到大排序，原数组也被升序排序了

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

sort排序方法

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
        }
        else if (value1 == value2) {
                return 0;
        }
        else {
                return 1;   //返回1, 表示会交换值
        }
}
var box = [0, 1, 10, 5, 15];
var aseBox = box.sort(compare);   //返回升序的数组
```

操作数组的方法

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 box = [2, 3, 4, '绿箭侠', '黑寡妇'];
var box2 = box.concat('美队', '雷神');
alert(box);
alert(box2);
```

操作数组的方法

slice()：不修改原数组，将原数组中的指定区域数据提取出来

```
var box = [2, 3, 4, "绿巨人"];
var box2 = box.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)
alert(box);
alert(box2);
```

操作数组的方法

splice(): 截取原数组中指定的数据, 会改变原数组

```
var box = [2, 3, 4, '绿箭侠', '黑寡妇'];
var box2 = box.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
alert(box);     //[2, 3, '绿箭侠', '黑寡妇'];
alert(box2);  //4
```

操作数组的方法

splice(): 截取原数组中指定的数据, 会改变原数组

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

```
box.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

```
box.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

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

- **用数组的元素组成字符串 – join()**

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

作业

必做：

1，把课堂上的案例照着敲一遍

2，冒泡排序自己写一遍

3，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:4,[1,3,4,2]

4，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

5. **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

第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

作 业

■ 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”。