

二维数组

- 在严格意思上面来讲，JavaScript没有二维数组的概念，它只有多维数组的概念
- 二维数组指的是当一个数组当中的元素又是一个数组的时候，它就变在了二维数组

二维的数组的定义

```
1 var teachers = ["标哥","飞哥","桃哥"];           //将老师放在了一个数组里
2 var stus = ["颜一鸣","李心悦","李康"];           //将学生也放在了一个数组里
3 var person = [teachers,stus];                     //这个时候数组里面的元素又是一个数组
```

上面的 `person` 当中的 `teachers` 以及 `stus` 都是一个数组，这种情况我们就把 `person` 叫二维数组

```
1 var arr1 = new Array(3);
2 var arr2 = new Array(3);
3 var arr = new Array(arr1,arr2);
```

上面的代码还可以简化成下面的写法

```
1 var arr = new Array(new Array(3),new Array(3));
```

还可以直接静态初始化

```
1 var arr = new Array(new Array("a","b","c"),new Array("d","e","f"));
```

我们之前在学习一维数组的时候我们也知道可以使用 `[]` 来创建数组，所以上面的代码可以直接写化成下面的写法

```
1 var arr = [
2     ["a","b","c"],
3     ["d","e","f"]
4 ];
```

二维数组的定义方式是有多多种多样的，只需要参照原来的一维数组进行就可以了

二维数组的特殊情况

首先我们先要弄清楚一，JS当中的数组与其它语言的数是不一样

1. 不限定数据类型
2. 不限定数组的长度

正是因为有了上面这两种特殊情况，所以我们先来看一下下面的几种方式的二维数组

第一种特殊情况

```
1 var arr = [  
2     ["颜一鸣", "李心悦", "李康"],  
3     ["陈韩家", "许才奇", "何祥宇", "张恒"]  
4 ];
```

上面的代码是合法的，因为JS的数组没有限定长度

第二种特殊情况

```
1 var arr =[  
2     ["颜一鸣", "李心悦", "李康"],  
3     "陈韩家",  
4     10  
5 ]
```

上面的代码也是合法的，因为JS的数组没有固定数据类型，里面的元素可以是任何数据类型

第三种特殊情况

```
1 var arr = [  
2     ["颜一鸣", "李心悦", "李康"],  
3     [  
4         ["标哥", "飞哥"],  
5         ["桃子"]  
6     ],  
7     "陈韩家",  
8     10  
9 ]
```

上面的代码也是合法的，因为JS没有长度的限制，也没有数据类型的限制

二维数组的取值与赋值

在之前讲一维数组的时候我们讲到过数组的取值是通过下标（索引）来完成的，同理，二维数组也是通过下标来完成的

```
1 var arr = [  
2     ["颜一鸣", "李心悦", "李康", "贺锐"],  
3     ["陈韩家", "许才奇", "何祥宇", "张恒"]  
4 ];
```

我们现在先在内存当中看一下这一个二维数组是如何表现

行 \ 列	0	1	2	3
0	颜一鸣	李心悦	李康	贺锐
1	陈韩家	许才奇	何祥宇	张恒

二维数组的取值可以像表一样去操作

```
1 var row0 = arr[0];           //这相当于取了第0行的数据  
2 var s1 = row0[1];           //这相当于取这一行的第1个数取
```

上在的操作是一步一步进行的，现在合起来一起进行

```
1 var s1 = arr[0][1];           //李心悦  
2 var s2 = arr[1][2];           //何祥宇
```

赋值也是一样的，现在把张恒换成“标哥哥”

```
1 arr[1][3] = "标哥哥";
```

总结：多维数组的取值一层一层向下找就可以了

```
1  var arr = [  
2      ["颜一鸣", "李心悦", "李康"],  
3      [  
4          ["标哥", "飞哥"],  
5          ["桃子"]  
6      ],  
7      "陈韩家",  
8      10  
9  ];
```

在上面的数组里面，我们要找出"飞哥"

```
1  arr[1][0][1];
```

二维数组的应用点

二维数组在日常开发与工作当中用于展示表的信息是最好的，同时也是用于展示平面信息是最好的

操作说明：玩家1：wasd上左下右，space射击；玩家2：方向键，



上面的游戏地图就是一个面面信息，这个地图我们可以使用二维数组的方式去完成

[illegible]

同时，电影购票里面的座位信息也是可以使用二维数组去完成的




```

11     if (Array.isArray(item)) {
12         item.forEach(function (item2) {
13             console.log(item2);
14         });
15     }
16     else {
17         console.log(item);
18     }
19     //如果不是数组就直接打印
20 });

```

3. 现在有以下多维数组，请将下面的多维数组中每一个元素遍历出来打印

```

1  var arr = [
2      ["颜一鸣", "李心悦", ["李康", "贺锐"]],
3      ["陈韩家", ["许才奇", "何祥宇"], "张恒"],
4      "陈韩家",
5      [10, [20, [30, 40]]]
6  ];

```

上面的题目很明显是一个多维数组了，我们不能再手动地去打印了，要考虑另外的办法

提示：可以使用递归

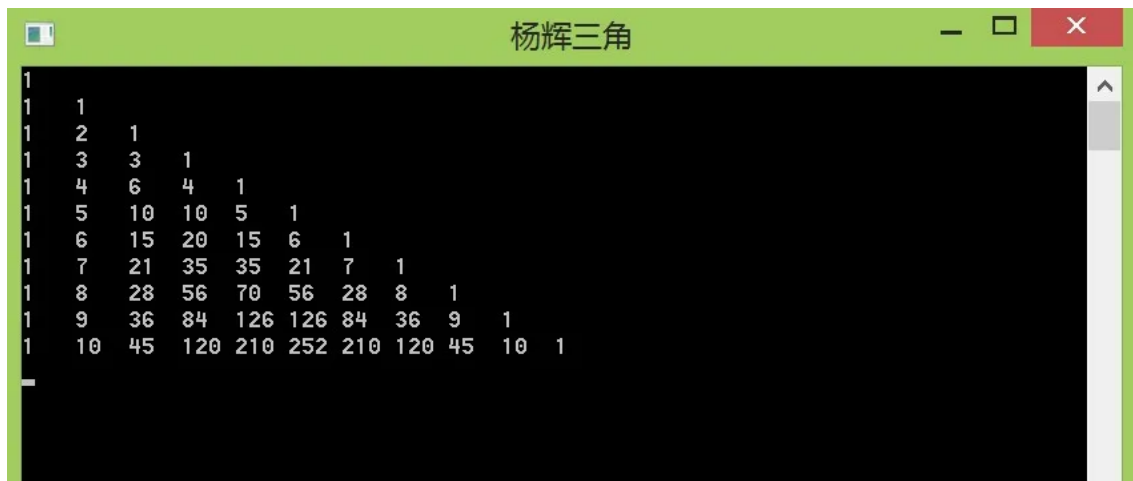
```

1  function printArray(aaa){
2      if(Array.isArray(aaa)){
3          //首先就判断你是不是数组
4          aaa.forEach(function(item){
5              printArray(item);
6          })
7      }
8      else{
9          console.log(aaa);
10     }
11 }

```

4. 打印杨辉三角的前10行

提示：使用二维数组去完成



5. 100个人手拉手围成一个圆，开始数数，数到3的倍数的人退出，最后剩下一个人的是之前的第几个人【约瑟夫环】

提示：使用一维数组完成

```
1 // 这里是100个人，我们创建一个长度为100的数组，每个人最初都用0表示
2 //第一步：创建长度为100的数组
3 var arr = new Array(100);
4 //第二步：在长度为100的数组里面全部放0，代表人
5 for (var i = 0; i < arr.length; i++) {
6     arr[i] = 0;
7 }
8 //第三步：开始数数，数到3的倍数的人退出
9 //如果这个人数到3了，我们就把这个人的0换成1，1就表示这个人已经退出了
10 var count = 0;
11 while (true) {
12     for (var i = 0; i < arr.length; i++) {
13         if (arr[i] == 1) {
14             //说明这个已经退出了，不用喊了
15             continue;
16         }
17         count++;
18         if (count == 3) {
19             arr[i] = 1;
20             count = 0;
21         }
22     }
23     //在什么情况下它就不数123了，只有剩下唯一的1的个人，用0表示，
24     if (arr.indexOf(0) == arr.lastIndexOf(0)) {
25         break;
26     }
27 }
28
29 //第四步：最终理想的状态就是arr数组里面只剩下1个0
30 //说明当前这个数组arr里面只能1个0
31 console.log(arr.indexOf(0) + 1);
```