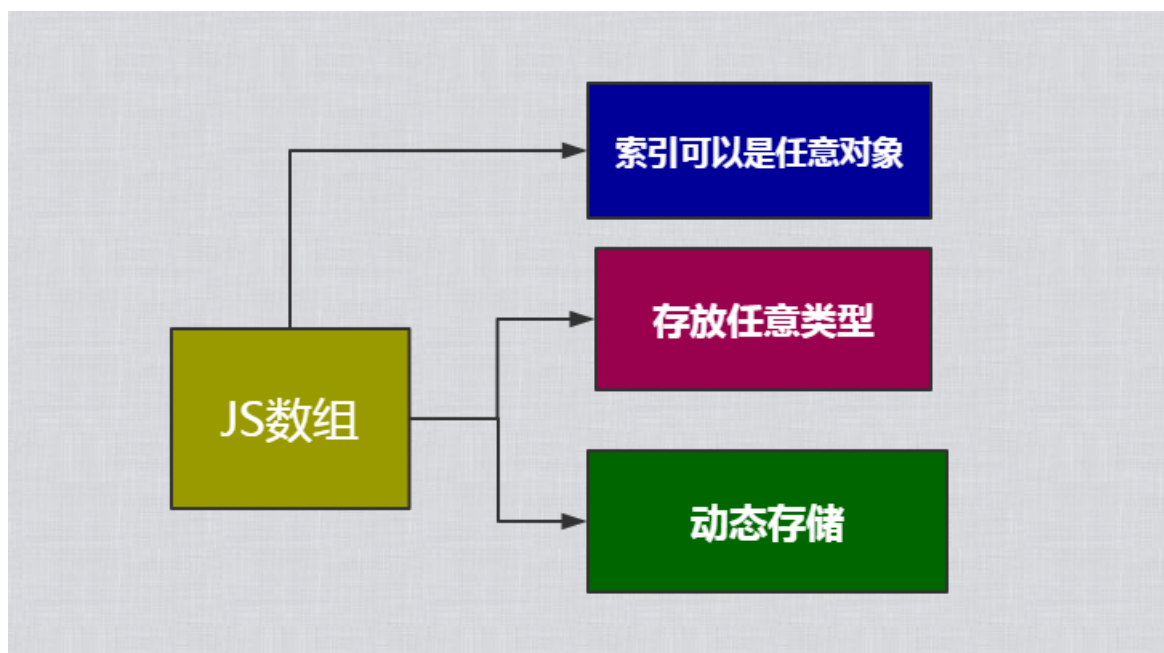


## 小兔JS教程（四）-- 彻底攻略JS数组



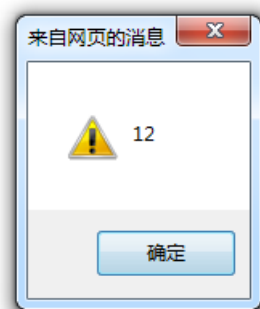
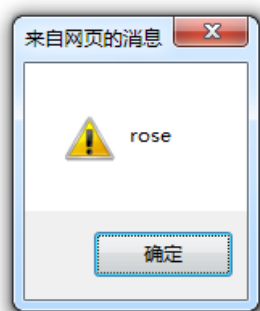
在开始本章之前，先给出上一节的答案，参考答案地址：

<http://www.xiaotublog.com/demo.html?path=homework/03/index2>

### 1.JS数组的三大特性

在JavaScript中，数组和其他编程语言中的数组有所不同。首先体现在，在一般的编程语言中，数组的索引只能是数字，而js数组中，它的索引可以是任意类型。比如，你可能会看到这样的代码：

```
var arr = []; //新建一个数组  
arr['name'] = 'rose';  
arr['age'] = 12;  
alert(arr['name']);  
alert(arr['age']);
```



这个例子说明，在js中，数组的索引不一定只能是数字，但是，如果用数字作为索引的话，效率无疑会比较高一点，它内部必然做了优化。Js数组除了这一点比较特殊之外，还有一个需要注意的地方，就是它可以是动态的。我们知道，在Java中，要使用数组，必须要先对其进行初始化，也就是说，我得先规定好，这个数组的容量是多大，然后，我才能使用这个数组。在使用

的过程中，要时刻注意好，不能超出数组的容量。如果数组要扩容，不管怎样，我都只好重新申请一块空间，new一个新的数组，然后把老数组中的那部分，嫁接拷贝到新的数组里面去。这就是Java数组的扩容概念。而在js中，数组的扩容变得毫不费力，因为它本来就是动态的。所以，你往往会看到这样的代码：

```
var arr = []; //新建一个数组,长度为0

arr[0] = 100; //动态扩容，现在长度为1

arr[1] = 200; //动态扩容，现在长度为2

alert(arr.length);
```



很好用吧，在这一点上，js数组是非常灵活的。最后，js数组就像一个篮子，随便你放什么都可以。小到一个数字，字符串，大到对象，函数，随便你放什么，都可以的。

```
var arr = []; //新建一个数组,长度为0

arr[0] = 100; //动态扩容，现在长度为1

arr[1] = 200; //动态扩容，现在长度为2

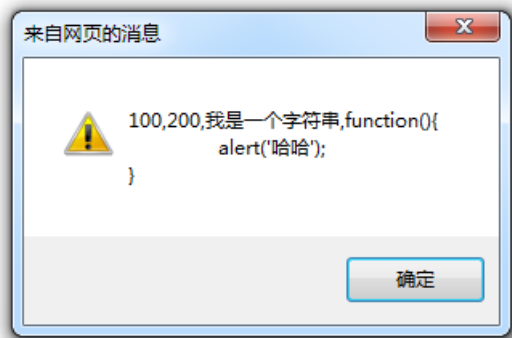
arr[2] = "我是一个字符串";

arr[3] = function(){

    alert('哈哈');

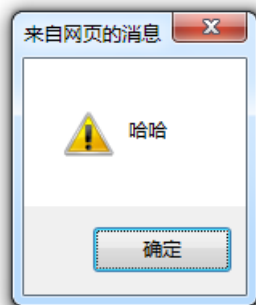
};

alert(arr);
```



在索引为3的地方，我们放入了一个函数，还是那句话，我现在问，这个函数有没有打括号，是不是还没有。好的，既然没有打括号，那么这个函数就不会被执行。也就是说，这个函数目前来看，就是和数字，字符串差不多的东西，还没有什么攻击力！我们可以这样去调用它：

```
arr[3]();
```



这样应该很好理解吧。当然了，这里我不太愿意像背字典一样，把每一个细节都讲一遍，我觉得那样没有太大意义。就算现在你记住了，过一段时间你说不定又忘记了。还是那句话，通过不断地写，不断地写，持续不断地写项目，在项目中把这些知识点练起来，这才是你真正可以获得的。

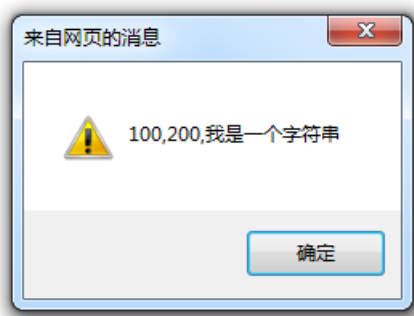
## 2.给数组赋值

在上面的例子中，我们可以通过这样的方式给数组赋值：

```
var arr = []; //新建一个数组,长度为0  
  
arr[0] = 100; //动态扩容，现在长度为1  
  
arr[1] = 200; //动态扩容，现在长度为2  
  
arr[2] = "我是一个字符串";
```

这就是给数组赋值的第一种方式，通过数组扩容的方式，即刻赋值。其实还有另一种方式，就是通过push方法来给数组赋值。我们用这种方式把上面的代码重写一遍，无非就是这样：

```
var arr = []; //新建一个数组,长度为0  
  
arr.push(100); //动态扩容，现在长度为1  
  
arr.push(200); //动态扩容，现在长度为2  
  
arr.push("我是一个字符串");  
  
alert(arr);
```



这样是不是也可以呢？好，现在我就想要请问一下了，push是什么。你不要把它想得太复杂了，我就问你，它后面有没有括号，有，好的。什么东西可以打括号？是不是方法呀，只有方法才可以打括号，打了括号就是去把它执行掉了。接下来再问，什么东西可以调用方法，是不是对象呀？OK，那么我们写的这个：

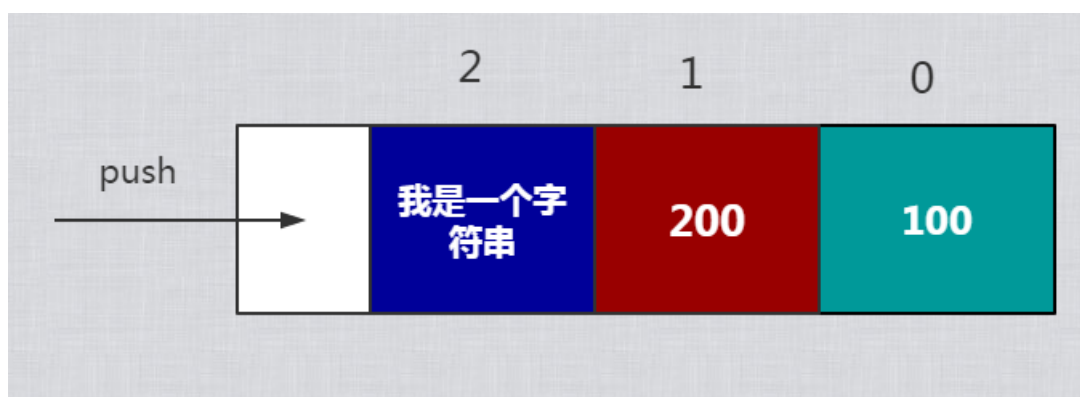
```
var arr = [];
```

这个arr是什么，是不是一个对象呀？好的，理解了这一点就好办多了。其实在JS中，数组其实有一个构造函数的。他就是Array，所以，上面这句代码，其实是这么一回事：

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

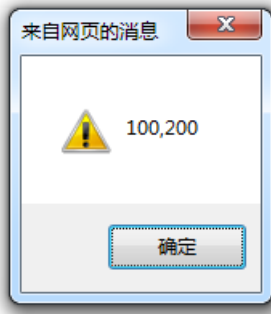
Array是js自带的一个函数，它是一个构造函数，相当于Java中类的概念。我们可以通过new一个函数的方式，来获得一个实实在在的对象。在这个例子中，arr就是一个实实在在的对象了。我再强调一遍，不要把简单的东西弄复杂了，如果你脑袋里只想着复杂，那么就永远看不到简单。只有对象才可以调用属性和方法。在这个例子中，arr固然是一个对象，所以，它可以调用一个叫做push的方法。

push的意思就是动态地给数组扩容一个单位，并且把对应的值放进去，就这么简单。



有了塞东西进去的方法，当然还有把东西拿出来方法， 没错就是pop。

```
var arr = new Array(); //新建一个数组,长度为0  
arr.push(100); //动态扩容, 现在长度为1  
arr.push(200); //动态扩容, 现在长度为2  
arr.push("我是一个字符串");  
arr.pop();  
alert(arr);
```



pop可以理解为弹出，弹出的就是最后一次push进去的值。

### 3.数组常用方法

接下来，介绍一些数组的常用方法。

#### 3.1 如何复制一个数组

为什么要复制数组呢？因为在实际项目中，你可能会遇到这样的问题。我拿到一个数组，需要临时保存下来，怎么办？你可能会写这样的代码：

```
var arr = new Array(); //新建一个数组,长度为0  
arr.push(100); //动态扩容, 现在长度为1  
arr.push(200); //动态扩容, 现在长度为2  
var arr2 = arr;  
arr.pop();  
arr.pop();  
alert(arr2);
```

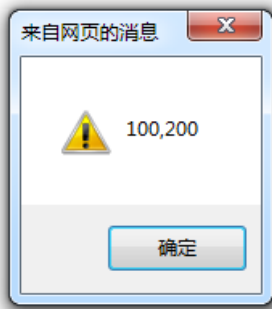
你希望通过arr2来把arr的数据保存下来，结果，一旦arr改变了，arr2也跟着改变了。



不好意思，没有。这是因为arr2和arr都是指向了同一块内存区间，一个改变，另一个也就跟着改变了。那怎么办呢？用slice就可以啦。

```
var arr = new Array(); //新建一个数组,长度为0  
arr.push(100); //动态扩容, 现在长度为1
```

```
arr.push(200); //动态扩容, 现在长度为2  
var arr2 = arr.slice();  
arr.pop();  
arr.pop();  
alert(arr2);
```



这样就可以了。

### 3.2 如何把数组转换成字符串

这个需求也经常会碰到，最常见的就是页面需要做一个批量操作，列表上勾选N条数据，一般来说，是拿到一个JSON数组。比如这样的：

```
var rows = [{id:1,name:'aaa'}, {id:2,name:'bbb'}];
```

现在，我要拿到这些id，通过ajax传递到后台，然后后台通过这些id去做某些操作。那换做是你，你打算怎么做呢？有一种办法，就是手动拼接这些id，用逗号分隔一下，做成一个字符串，然后传递到后台，后台再用split方法进行解析。

你可能会这么做：

```
var rows = [{id:1,name:'aaa'}, {id:2,name:'bbb'}];  
var ids = '';  
//拼接id  
for(var i=0;i<rows.length;i++){  
    if(i == rows.length - 1){  
        ids += rows[i].id  
    }else{  
        ids += rows[i].id + ',';  
    }  
}  
alert(ids);
```



这样做自然是可以的咯，可是，如果用数组的话，我们可以使用join方法。

```
var rows = [{id:1,name:'aaa'}, {id:2,name:'bbb'}];
```

```

var ids = [];

//拼接id
for(var i=0;i<rows.length;i++){

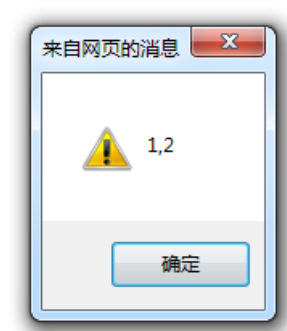
    ids.push(rows[i].id);

}

ids = ids.join(','); //转换成用逗号分隔的字符串

alert(ids);

```



### 3.3 如何去除数组中的重复元素

我这边封装了一个简单的方法，可以去重：

```

var arr = [1,2,3,4,5,1];

function unique(arr){

    //新建一个数组，用来存放原数组中所有不重复的数据

    var afterUnique = [];

    //新建一个JSON对象

    var obj = {};

    //定义一个常量

    var EXIST = 1;

    //循环这个数组

    for(var i=0;i<arr.length;i++){

        /*

        obj[arr[i]]的意思是在obj中，key为arr[i]的值

        如果不等于EXIST，表示不存在，那么就把该数据push到数组中

        同时，将这个数据标记为已存在 ==> obj[item] = EXIST;

        */

        if(obj[arr[i]] != EXIST){

            afterUnique.push(arr[i]);

            obj[arr[i]] = EXIST;

        }

    }

    return afterUnique;

}

alert(unique(arr));

```

当然，这个方法里面也用到了JSON。下一节，我们就来学习一下JSON吧。在本节中，有些地方我故意没有细讲，但凡是我觉

得可以通过自行百度来解决的问题，就没有必要在博客中写明了，我也推荐大家能够有目的地去百度一些资料，自己学习。我的博客只讲一些关键的点，以及很多我认为重要的东西，绝对不会像翻字典一样面面俱到，因为我觉得那样没意义，时间一长，还不是忘掉了。只有当你自己真的在项目中遇到了，才会真正的记住。我不希望把读者思维局限在我的文章里面，所以我只讲关键的，常用的。其他东西，留给大家自己去百度，自己去提升。把思路放开来，你会收获更多。

作业：<http://www.xiaotublog.com/demo.html?path=homework/04/index>