**数组：**

* 数组也是对象，它和我们普通对象类似，也是用来存储值的
* 数组用数字来操作索引的
* 数据类型也是对象
* 索引：从0开始的整数就是索引

**创建数组：**

* **语法**：数组[索引]=值 索引是由0开始

arr[0] = 10;

arr[1] = 10;

* **读取**： 数组[索引] = 值

arr[0];

**获取数组长度(length):**

* **语法：**数组.length arr.length;

对于连续的数组，使用length可以获取数组的长度

对于不连续的数组，使用length会获取数组的最大索引+1

尽量不要创建非连续的数组

**修改length**

arr.length如果修改的length大于原长度，则多出部分会空出来

如果修改的length小于原长度，则会多出的元素会被删除

向数组的最后一个位置添加元素arr[arr.length] = 70;

**字面量创建数组**

语法：[] 【var arr=[];】

使用字面量数组就能指定数组的元素

var arr=[1，2,3,4,5,6]索引从0开始

* arr=[10];创建一个数组中只有一个数10

console.log(arr[0]);

* arr = new Array(10);创建一个为10的数组

数组中的元素可以是任意类型，

* **可以是对象**

var obj = {name:’sunwuk’};

arr[arr.lenght] = obj;

console.log(arr[5]);

* **可以是函数**

arr=[function(){},function(){},function(){}]

打印函数;arr[0]();

* **可以放数组(二维数组)**

arr=[[1,2,3], [1,2,3], [1,2,3]]

**数组的方法：**

**push()：**想数组末尾加一个或多个元素，返回新长度

var result=arr.push(‘tangseng’,’zhizhuj’)

arr 是返回所有的数，resule是返回长度

**unshif():**想数组头头加一个或多个元素，返回新长度

向前边插入元素的时候，其他元素索引都想后

var result=arr. unshif (‘tangseng’,’zhizhuj’)

arr 是返回所有的数组，resule是返回长度

**shif():**该方法是删除数组的第一个元素，并将删除的元素作为删除值返回

var shif =arr.pop();//调用一次就删除一次，result=删除的值

**pop():**该方法是删除数组的最后一个元素，并将删除的元素作为删除值返回

var result =arr.pop();//用一次就删除一次末尾result=删除的值

**slice()**可以用来从数组提取指定元素该方法不会改变元素数组，而是将截取到的元素封装到一个新数组中返回  
 **参数：**  
 1.截取开始的位置的索引,包含开始索引  
 2.截取结束的位置的索引,不包含结束索引  
 第二个参数可以省略不写,此时会截取从开始索引往后的所有元素  
 索引可以传递一个负值，如果传递一个负值，则从后往前计算  
 1 倒数第一个  
 2 倒数第二个

arr = ["孙悟空","猪八戒","沙和尚","唐僧","白骨精"];

**截取**

var result = arr.slice(1,4);

console.log(result)

**//从第一个开始，到第四个之前，得到的值要给新数组或者变量**

result = arr.slice(3);  
console.log(result)

**//第三个数开始截取到末尾之间的值**

result = arr.slice(1,-2);

//第一个开始，到倒数第二个之前不包括的数，

**splice()**  
 可以用于删除数组中的指定元素  
  **使用splice()会影响到原数组，**会将指定元素从原数组中删除  
 并将被删除的元素作为返回值返回

**参数：** 第一个，表示开始位置的索引  
 第二个，表示删除的数量  
 第三个及以后。。

可以传递一些新的元素，这些元素将会自动插入到开始位置索引前边  
后面加入的是替换删除的位置添加新的东西

arr = ["孙悟空","猪八戒","沙和尚","唐僧","白骨精"];  
var result = arr.splice(3,0,"牛魔王","铁扇公主","红孩儿");//(删除的下标，删几个)  
console.log(arr);  
//console.log(result);

**concat()**可以连接两个或多个数组，并将新的数组返回  
 该方法不会对原数组产生影响

**join()**该方法可以将数组转换为一个字符串该方法,不会对原数组产生影响，而是将转换后的字符串作为结果返回

在join()中可以指定一个字符串作为参数，这个字符串将会成为数组中元素的连接符  
如果不指定连接符，则默认使用,作为连接符

**reverse()**

该方法用来反转数组（前边的去后边，后边的去前边）

该方法会直接修改原数组

**sort()**  
\* - 可以用来对数组中的元素进行排序  
\* - 也会影响原数组，默认会按照Unicode编码进行排序

即使对于纯数字的数组，使用sort()排序时，也会按照Unicode编码来排序，  
\* 所以对数字进排序时，可能会得到错误的结果。  
\*

我们可以自己来指定排序的规则

我们可以在sort()添加一个回调函数，来指定排序规则，  
 回调函数中需要定义两个形参,  
 浏览器将会分别使用数组中的元素作为实参去调用回调函数  
 使用哪个元素调用不确定，但是肯定的是在数组中a一定在b前边  
 浏览器会根据回调函数的返回值来决定元素的顺序，  
 如果返回一个大于0的值，则元素会交换位置  
 如果返回一个小于0的值，则元素位置不变  
 如果返回一个0，则认为两个元素相等，也不交换位置  
 如果需要升序排列，则返回 a-b  
 如果需要降序排列，则返回b-a

arr = [5,4,2,1,3,6,8,7];  
arr.sort(function(a,b){   
 //前边的大  
 /\*if(a > b){  
 return -1;  
 }else if(a < b){  
 return 1;  
 }else{  
 return 0;  
 }\*/  
 //升序排列  
 //return a - b;  
 //降序排列  
 return b - a; });  
console.log(arr);

**数组的遍历**

所有的数都访问，并且只访问一次

for（var I = 0; I<arr.length;i++）{//最大值是长度

console.log(arr[i]);//逐个输出数组

**例子：把符合条件的数组提取出来**

function person(name,age) {  
 this.name = name;  
 this.age = age;  
}**//构造函数，方法用下面的方法写  
person.prototype.toString=function () {**  
 return 'person[name='+this.name+'age='+this.age+']';  
};**//构造函数的方法，用toString打印出来，**

**//构造函数的实例化**var per = new person('zhu',18);  
var per1 = new person('zhu1',19);  
var per2 = new person('zhu2',17);  
var per3 = new person('zhu3',133);  
var per4 = new person('zhu4',14);  
**//定义一个函数来放实例**var arr1=[per,per1,per2,per3,per4];

function getAdult(arr) {  
  **//创建一个新数组** var arr2=[];  
  **//遍历参数数组** for(i = 0;i <arr.length;i++){  
  **//取出符合的数组** if(arr[i].age > 18){  
 arr2.push(arr[i]);**//放到新的数组的后面**} }  **return arr2;//返回数组，返回数组，返回数组**}

alert(getAdult(arr1));//调用函

**forEach()方法**

需要一个函数作为参数，你支持IE8以下  
 - 像这种函数，由我们创建但是不由我们调用的，我们称为回调函数  
 - 数组中有几个元素函数就会执行几次，每次执行时，浏览器会将遍历到的元素  
 - 以实参的形式传递进来，我们可以来定义形参，来读取这些内容  
 **浏览器会在回调函数中传递三个参数：** 第一个参数，就是当前正在遍历的元素  
 第二个参数，就是当前正在遍历的元素的索引

第三个参数，就是正在遍历的数组

**例子:**

var arr = ["孙悟空","猪八戒","沙和尚","唐僧","白骨精"];

arr.forEach(function(value , index , obj){  
 console.log(value);  
});

arr.forEach(fun);

例子：去重复：去除一组数组中的重复出现的元素。



