1. join（） 不创建数组，返回字符串。  
   如果不指定分隔符，则默认适应逗号分隔。
2. reverse（） 不创建数组，返回逆序的数组。  
   倒序
3. sort（） 不创建数组，返回排序后的数组。  
   排序。不带参数调用的话，数组以字母表的顺序排序。  
   传入一个函数，想让第一个参数在前，返回小于0的值。如果想让第一个参数在后，返回大于0的值。

concat（） 创建新数组并返回。

    var a=[1,2,3];

    a.concat(4,5); //[1,2,3,4,5]

    a.concat([4,5]); //[1,2,3,4,5]

    a.concat([4,[5,[6,7]]]) //[1,2,3,4,5,[7,7]]

1. slice（） 返回原数组的一个子数组，原数组不会改变。  
   包含第一个参数，但不包含第二个参数。正数索引从0开始，负数索引从-1开始表示倒数第一个。如果只指定一个参数，返回从该位置到数组末尾的所有元素。
2. splice（）删除元素或添加元素。改方法会改变原数组，一定是返回一个数组，返回的数组是被删除的元素组成的，如果没有删除元素，则返回空数组[]。  
   第一个参数表示从哪个地方开始删除数组，该位置也要被删除；或者说当添加元素的时候，该被添加的元素充当第几个元素。  
   a=[1,2,3,4,5];  
   a.splice(3,2); 返回[4,5],a变为[1,2,3]  
   a=[1,2,3,4,5];  
   a.splice(3,0,’a’); 返回[],a变为[1,2,3,’a’,4,5]  
   第二个参数表示删除的个数，如果是0就表示要添加了。如果省略第二个参数，则表示从该位置起所有的元素都会被删除掉。
3. push（）和pop（） 会改变原数组，返回新数组的长度。
4. shift（）和unshift（） 会改变原数组，返回新数组的长度。
5. toString（） 和join（）是一个效果
6. ES5的方法  
   ES5的数组方法通常接受一个函数，会对每个元素都调用一次这个函数。该函数一般有3个参数，分别为数组元素、数组索引和数组本身。
7. map（） 将数组的每个元素传递给指定的函数，并返回一个数组。  
   与forEach（）相同的是传递的函数相同，但是这个函数必须要有return返回值，不然得不到map的数组。Map方法不会修改原数组，它会创建新的数组。它返回的是和原数组一样长度的数组，具有相同的缺失元素。
8. filter() 原数组不会改变，返原数组的一个子集。  
   必须要有return语句，判断false则不加入返回的数组，true加入返回的数组。  
   filter（）会跳过稀疏数组中缺少的元素，如压缩数组：  
   a.filter(function(x){return true})
9. every()和some() 返回true或false  
   必须要有return语句
10. indexOf（）和lastIndexOf（）  
    找到第一个检索的索引位置，如果没找到就返回-1.lastIndexOf（）从末尾到头数。
11. reduce和reduceRight（）  
    这个数组方法传入的参数不一样。  
    必须要有return语句，用指定的函数将整个数组进行组合，最终生成单个值。第二个参数为可选值，如果有的话就是初始值。  
    a=[1,2,3,4];  
    sum=a.reduce(function(a,b){return a+b })
12. ES5中的Array.isArray()静态方法可以检测对象是否为数组。