1. 判断浏览器是否支持indexOf

* if (Array.prototype.indexOf)

1. 遍历(求和)

* forEach方法：

arr.forEach(element => {

s += element;

});

arr.forEach(function(a){

s += element;

});

* reduce方法：

arr.reduce(

function(prev,cur){

return prev + cur;

}

1. 删除数组中某个元素

* 去掉某段：splice: 原数组上操作，arr.splice(m,1); 删除从下标为m的元素开始的1个元素。
* 去掉某段：slice(start,end): 返回一个新数组，end可以省略。
* 去尾：arr.pop()

注意：是在原数组上进行的修改。

1. 复制数组

* arr.slice(0);
* 遍历数组
* Arr.concat()

1. 添加元素

* 头插：arr.unshift( item )
* 尾插：arr.push( item )
* 插入某个位置：arr.splice(插入点位置, 0, items)

1. 数组合并

* a.concat(b) 返回新的数组，性能高
* slice(0)+push

1. 找重复元素

* 两个数组，一个用于存储最终得到的重复元素，另一个用于记录每个元素出现的次数（下标为arr中的值）。

1. 转为整数

* 举例，如果 string 以 "0x" 开头，parseInt() 会把 string 的其余部分解析为十六进制的整数。如果 string 以 0 开头，那么 ECMAScript v3 允许 parseInt() 的一个实现把其后的字符解析为八进制或十六进制的数字。如果 string 以 1 ~ 9 的数字开头，parseInt() 将把它解析为十进制的整数。
* return parseInt(num,10);

1. 计时器

* setTimeout( funciton(){}, 100)：以100毫秒每次的速度执行起里面定义的函数。

1. 判断参数是否为空

* num == 'null'（等同于undefined ,null）
* a == "" || a == null || a == undefined（"",null,undefined）
* !a（"",null,undefined,NaN）

1. 类型判断

* typeof(xxx)==’number’

1. 函数传参

* 将数组中值，作为函数的参数：fn.apply(this, arr)
* 输入：

function (greeting, name, punctuation) {return greeting + ', ' + name + (punctuation || '!');}, ['Hello', 'Ellie', '!']

输出：Hello, Ellie!

1. 更改函数的上下文

* 用obj对象替换函数体：fn.apply(this, obj)
* 输入：

function () {return this.greeting + ', ' + this.name + '!!!';}, {greeting: 'Hello', name: 'Rebecca'}

输出：Hello, Rebecca!!!

1. 删除json对象数组中的某个元素

* Marr.splice(i,1)
* 举例：[{‘id’:123,’name’:’lily’}, {‘id’:12,’name’:’tom’}]