1. 博客
2. 检测数组
   1. instanceof

console.log([1,2] instanceof Array);//true

当网页中包含多个框架时，这个方法不行

* 1. toString

console.log(Object.prototype.toString.call(arr)==='[object Array]');//true

* 1. Array.isArray()

console.log(Array.isArray(arr));//true

1. 数组与字符串
   1. 可索引

var arr=[1,2,3];

var str='abc';

console.log(str[0]);//a

console.log(arr[0]);//1

* 1. 转换

console.log(str.split());//["abc"]

console.log(arr.join());//1,2,3

* 1. 拼接

console.log(arr.concat([4]));//[1, 2, 3, 4]

console.log(str.concat("d"));//abcd

* 1. 提取子数组/字符串

console.log(arr.slice(1));// [2, 3]

console.log(str.slice(1));//bc

* 1. 位置

console.log(str.indexOf('a'),str.lastIndexOf('a'));//0 0

console.log(arr.indexOf(2),arr.lastIndexOf(2));//1 1

* 1. 顺序

arr.reverse() arr.sort()

str没有这2个方法，只能用apply或call

console.log(Array.prototype.reverse.apply(str.split("")));// ["c", "b", "a"]

1. reduce

reduce方法接收一个函数作为累加器，第二个参数(可选)为初始值

此函数有4个参数：上一次调用返回值或初始值、当前值、当前索引、arr数组

reduce方法的返回类型和初始值相同

应用：

求和

Array.prototype.sum=function(){

return this.reduce(function(prev,cur){

return prev+cur;

})

}

console.log([1,1,1].sum());//3

由于reduce方法依次处理每个元素，所以可用来搜索某个元素，比如找出长度最长的数组元素

var arr=[1,2,'ab',45678];

function findLongest(entries){

return entries.reduce(function(prev,cur){

return cur.length>prev.length?cur:prev;

},'')

}

console.log(findLongest(arr));//'ab'

扁平化二维数组：

var matrix=[[1,2],[3,4]];

var flatten=matrix.reduce(function(prev,cur){

return prev.concat(cur);

})

console.log(flatten);[1,2,3,4]

在空数组上，不带初始参数调用reduce会导致类型错误异常。

1. map

map对数组的每一项运行给定函数，返回每次函数调用的结果组成的数组

map中函数的参数：当前元素、当前索引、当前数组

map还可以接收第2个参数，表示回调函数执行时this的指向

1. forEach

forEach方法对数组中的每一项运行给定函数，没有返回值

forEach还可以接收第2个参数，表示回调函数执行时this的指向

多层循环的this指向是不确定的，可以使用forEach的第二个参数固定this

var obj={

name:'cdx',

times:[1,2,3],

print(){

this.times.forEach(function(n){

console.log(this);//此this指向obj

},this)

}

}

obj.print();