数组API

一日三省且温故知新

1.数组转字符串

1.1. String(arr)： 将arr中每个元素转为字符串，用逗号分隔

固定套路: 对数组拍照: 用于鉴别是否数组被修改过

1.2. arr.join(“连接符”)： 将arr中每个元素转为字符串，用自定义的连接符分隔

```
//将字符拼接为单词
var chars=["H","e","l","l","o"];
console.log(chars.join(""));              //Hello
```

1.数组转字符串

固定套路:

1.将字符组成单词: chars.join("")->无缝拼接

扩展: 判断数组是空数组: arr.join("")==""

2.将单词组成句子: words.join(" ")

3.将数组转化为页面元素的内容:

```
"<开始标签>"+
arr.join("</结束标签><开始标签>")
+"</结束标签>"
```

2. 拼接和选取

不直接修改原数组，而返回新数组！

拼接: concat() 拼接两个或更多的数组，并返回结果

var newArr=arr1.concat(值1,值2,arr2,值3,...)

将值1,值2和arr2中每个元素,以及值3都拼接到arr1的元素之后，返回新数组

其中: arr2的元素会被先*打散*，再拼接

```
var arr1 = [90, 91, 92];
var arr2 = [80, 81];
var arr3 = [70, 71, 72,73];
var arr4 = arr1.concat(50, 60, arr2, arr3);

console.log( arr1 );                    //现有数组值不变
console.log( arr4 );
```

2. 拼接和选取

不直接修改原数组，而返回新数组！

选取: slice() 返回现有数组的一个子数组

var subArr=arr.slice(starti,endi+1)

选取arr中starti位置开始，到endi结束的所有元素组成新数组返回——原数组保持不变

强调: 凡是两个参数都是下标的函数，都有一个特性:含头不含尾

```
var arr1 = [10, 20, 30, 40, 50];
var arr2 = arr1.slice(1, 4);          //20,30,40
var arr3 = arr1.slice(2);             //30,40,50
var arr4 = arr1.slice(-4, -2);        //20,30

console.log(arr1);                    //现有数组元素不变
```

2. 拼接和选取

选取简写:

1. 一直选取到结尾: 可省略第二个参数

2. 如果选取的元素离结尾近: 可用倒数下标:

arr.slice(arr.length-n,arr.length-m+1)

可简写为:arr.slice(-n,-m+1);

3. 复制数组:

arr.slice(0,arr.length);

可简写为:arr.slice();

3. 修改数组

删除: splice 直接修改原数组

arr.splice(starti,n);

删除arr中starti位置开始的n个元素不考虑含头不含尾

其实: var deletes=arr.splice(starti,n);

返回值deletes保存了被删除的元素组成的临时数组

```
var arr1 = [10, 20, 30, 40, 50];
var arr2 = arr1.splice(2, 1);
//var arr2 = arr1.splice(2, 2, 21,22,23);
//var arr2 = arr1.splice(2, 2, [91,92,93]);

console.log( arr1 );
console.log( arr2 );
```

3. 修改数组

插入:

arr.splice(starti,0,值1,值2,...)
在arr中starti位置，插入新值1,值2,...原starti位置的值及其之后的值被向后顺移

替换:

其实就是删除旧的，插入新的
arr.splice(starti,n,值1,值2,...)
先删除arr中starti位置的n个值，再在starti位置插入新值
强调: 删除的元素个数和插入的新元素个数不必一致。

4.颠倒数组

颠倒: reverse() 颠倒数组中元素的顺序

arr.reverse();

```
var arr1 = [10, 20, 30, 40, 50];
arr1.reverse();

console.log( arr1 );
```

强调: 仅负责原样颠倒数组，不负责排序

5. 排序

将元素按从小到大的顺序重新排列

排序API:

arr.sort(): 默认将所有元素转为字符串再排列

问题: 只能排列字符串类型的元素

解决: 使用自定义比较器函数

排序算法:

(手写) 冒泡 快速 插入排序

小结

数组API

- 数组转字符串
- 拼接和选取
- 颠倒数组
- 排序