

## 规范解读

- 1、会改变原数组
- 2、返回删除的数组
- 3、可单独添加元素
- 4、可单独删除元素
- 5、可添加元素&&删除元素

- 1、 Let O be the result of calling ToObject passing the this value as the argument.
- 2、 Let A be a new array created as if by the expression new Array() where Array is the standard built-in constructor with that name.
- 3、 Let lenVal be the result of calling the [[Get]] internal method of O with argument "length".
- 4、 Let len be ToUint32(lenVal).
- 5、 Let relativeStart be ToInteger(start).
- 6、 If relativeStart is negative, let actualStart be max((len + relativeStart),0); else let actualStart be min(relativeStart, len).
- 7、 Let actualDeleteCount be min(max(ToInteger(deleteCount),0), len – actualStart).
- 8、 Let k be 0.
- 9、 Repeat, while k < actualDeleteCount 将所有删除的元素添加到一个新的数组中 最后将其返回
  - a: Let from be ToString(actualStart+k).
  - b: Let fromPresent be the result of calling the [[HasProperty]] internal method of O with argument from.
  - c: If fromPresent is true, then
    - i: Let fromValue be the result of calling the [[Get]] internal method of O with argument from.
    - ii: Call the [[DefineOwnProperty]] internal method of A with arguments ToString(k), Property Descriptor {[[Value]]: fromValue, [[Writable]]: true, [[Enumerable]]: true, [[Configurable]]: true}, and false.
    - d: Increment k by 1.
- 10、 Let items be an internal List whose elements are, in left to right order, the portion of the actual argument list starting with item1. The list will be empty if no such items are present.
- 11、 Let itemCount be the number of elements in items.
- 12、 If itemCount < actualDeleteCount, then
  - a: Let k be actualStart.
  - b: Repeat, while k < (len – actualDeleteCount)
    - i: Let from be ToString(k+actualDeleteCount).
    - ii: Let to be ToString(k+itemCount).
    - iii: Let fromPresent be the result of calling the [[HasProperty]] internal method of O with argument from. 将from作为参数传递给O的内建方法[[HasProperty]] 结果赋值给fromPresent
    - iv: If fromPresent is true, then
      - 1、 Let fromValue be the result of calling the [[Get]] internal method of O with argument from.
      - 2、 Call the [[Put]] internal method of O with arguments to, fromValue, and true.
    - v: Else, fromPresent is false
      - 1、 Call the [[Delete]] internal method of O with arguments to and true.
    - vi: Increase k by 1.
  - c: Let k be len.

d: Repeat, while  $k > (\text{len} - \text{actualDeleteCount} + \text{itemCount})$

i: Call the `[[Delete]]` internal method of `O` with arguments `ToString(k-1)` and `true`. 调用O内建的方法`[[Delete]]`来删除 $k-1$ 位置上的元素

ii: Decrease  $k$  by 1.

13、 Else if  $\text{itemCount} > \text{actualDeleteCount}$ , then

a: Let  $k$  be  $(\text{len} - \text{actualDeleteCount})$ .

b: Repeat, while  $k > \text{actualStart}$

i: Let `from` be `ToString(k + actualDeleteCount - 1)`.

ii: Let `to` be `ToString(k + itemCount - 1)`

iii: Let `fromPresent` be the result of calling the `[[HasProperty]]` internal method of `O` with argument `from`.

iv: If `fromPresent` is true, then

1、 Let `fromValue` be the result of calling the `[[Get]]` internal method of `O` with argument `from`.

2、 Call the `[[Put]]` internal method of `O` with arguments `to`, `fromValue`, and `true`.

v: Else, `fromPresent` is false

1、 Call the `[[Delete]]` internal method of `O` with argument `to` and `true`.

vi: Decrease  $k$  by 1.

14、 Let  $k$  be `actualStart`.

15、 Repeat, while `items` is not empty

a: Remove the first element from `items` and let `E` be the value of that element.

b: Call the `[[Put]]` internal method of `O` with arguments `ToString(k)`, `E`, and `true`.

c: Increase  $k$  by 1.

16、 Call the `[[Put]]` internal method of `O` with arguments `"length"`,  $(\text{len} - \text{actualDeleteCount} + \text{itemCount})$ , and `true`.

17、 Return `A`.

## 实现思路

- 1、删除或添加元素时：没有涉及到的元素先进行位置的移动
- 2、接着再将需要添加的元素依次push进数组

## 源码实现

```
function mySplice(arr, start, deleteCount, ...item) {
  const O = arr;
  const A = new Array();
  const lenVal = O.length;
  const len = lenVal >>> 0;
  const relativeStart = Number(start);
  const actualStart = relativeStart < 0 ? Math.max(len + relativeStart, 0) :
Math.min(relativeStart, len); // 当start为负数是 从右边算起 最右边为-1
  const actualDeleteCount = Math.min(Math.max(Number(deleteCount), 0), len -
actualStart); // 删除的个数不能超过后面数组剩余的个数
  let k = 0;
  // 获取删除的数组
  while(k < actualDeleteCount) {
    const from = String(actualStart + k);
    // if(from in O) {
    //   const fromValue = O[Number(from)];
```

```

    // A.push(fromValue);
    // }
    const fromPresent = O.hasOwnProperty(from);
    if(fromPresent) {
        const fromValue = O[from];
        Object.defineProperty(A, k, {
            writable: true,
            enumerable: true,
            configurable: true,
            value: fromValue
        })
    }
    k++;
}
let items = [];
if(item.length > 0) {
    items = item;
}
const itemCount = items.length;
// 添加的个数小于删除的个数 这时候原有的元素要往左移
if(itemCount < actualDeleteCount) {
    let k = actualStart;
    while(k < (len - actualDeleteCount)) {
        const from = String(k + actualDeleteCount);
        const to = String(k + itemCount);
        const fromPresent = O.hasOwnProperty(from);
        if(fromPresent) {
            const fromValue = O[from];
            O[to] = fromValue;
            delete O[from];
        } else {
            // delete方法将某个位置上的值置为undefined
            delete O[to];
        }
        k++;
    }
}
// 添加的个数大于删除的个数 这时候原有的元素要往右移
// 移动元素时必须从数组最后一个开始移动 这样不会干扰到别的元素的移动
if(itemCount > actualDeleteCount) {
    let k = len - actualDeleteCount;
    while(k > actualStart) {
        const from = String(k + actualDeleteCount - 1);
        const to = String(k + itemCount - 1);
        const fromPresent = O.hasOwnProperty(from);
        if(fromPresent) {
            const fromValue = O[from];
            O[to] = fromValue;
        } else {
            delete O[to];
        }
        k--;
    }
}
let i = actualStart;
// 按队列顺序插入数据
while(items.length > 0) {
    const E = items.shift();

```

```
    o[String(i)] = E;
    i++;
  }
  o['length'] = len - actualDeleteCount + itemCount;
  return A;
}
```

参考链接:

- [splice原理分析](#)
- [官方规范](#)