

数组的一些方法

```
let arr = [1, 2, 3, 2, 1];
```

```
// 一 forEach => 遍历数组
```

```
// 二 map => 使用一个数组，利用某规则映射得到一个新数组
```

```
let mapArr = arr.map((v, i) => {  
    return v * v;  
});
```

```
arr.map((v, i) => v * v); // 如果只有一句话，可以省略大括号和return
```

```
console.log(mapArr); // [1, 4, 9, 4, 1]
```

```
// 三 filter => 找出符合条件的元素
```

```
// 3.1 找出数组中的偶数
```

```
let filterArr1 = arr.filter((v, i) => v % 2 == 0);  
console.log(filterArr1);
```

```
// 3.2 删除数组中的偶数
```

```
let filterArr2 = arr.filter((v, i) => v % 2 != 0); // 找出数组中的奇数  
留下来即可
```

```
console.log(filterArr2);
```

```
// 四 every => 判断数组中的所有元素是否都符合某种条件
```

```
// 4.1 数组中的数字是否都是偶数
```

```
let result1 = arr.every((v, i) => v % 2 == 0) // 所有的结果都为true，结  
果为true，有一个false，结果为false
```

```
console.log(result1)
```

```
// 4.2 数组中的数字是否都大于0
```

```
let result2 = arr.every(v => v > 0)
```

```
console.log(result2)
```

```
// 五 some => 判断数组中是否含有符合条件的元素
```

```
// 5.1 数组中是否有大于2的数
```

```
let result3 = arr.some(v => v > 2) // 只要有一个true, 结果为true  
console.log(result3)
```

```
// 5.2 数组中是否有小于1的数
```

```
let result4 = arr.some(v => v < 1)  
console.log(result4)
```