

**every ()** 判断数组是否满足 某个条件返回布尔值 会循环每一个值

```
1 //判断数组是否满足 某个条件返回布尔值 会循环每一个
2 console.log(arr.every(item => item > 0));
```

**some ()** 判断数组是否满足某个条件只要发现一个 就停止遍历//返回一个布尔值

```
1 //some 判断数组是否满足某个条件只要发现一个 就停止遍历
2 console.log(arr.some(item => item === 6));
```

**find**//查找数组元素满足某个条件的第一个值，只要找到了就停止，没有找到返回undefined

```
1 console.log(arr.find(item => item > 6))
```

**find**//查找数组元素满足某个条件的第一个索引，只要找到了就停止，没有找到返回-1

```
1 console.log(arr.findIndex(item=>item>6))
```

**forEach ()** 遍历数组 和for一样 但是无法停止除非抛出错误

```
1 //foreach,就普通循环遍历 注意没法停止循环 除非抛出错误
2 // console.log(arr.forEach(item))
```

**Array.from()** 将为数组转为真数组

```
1 ///Array.from()将伪数组转伪真数组
2 Array.from(btns)
```

**includes()**判断数组中是否包含这个元素

```
1 //includes 判断是否包含此元素
2 arr.includes(6)
```

**map ()** 创建一个新数组，这个数组的每个元素都调用一次提供函数后的返回值组成

```
1 arr.map(item => `♥${dd}`)
```

**flat()** // 扁平化数组

```
1 arr.flat(
```

## 字符串

**str.includes ()** 是否包含某个元素

```
1 let str='ll'
```

```
2 console.log(str.includes('l'));
```

判断字符串是否以某个字符开头

```
1 let str='ll'  
2 console.log(str.startsWith('l'));
```

## for of循环

可以循环值 包括 数组 字符串 map

```
1 let val=['hao','kk']  
2 for (const v of val) {  
3     console.log(v);  
4 }
```