

Set 和 Array 的区别

- API 不同
- Set 的成员不能重复，Array 的成员可以重复
- Set 是无序结构，即元素的插入顺序和遍历顺序不一定相同，Array 是有序结构。Set 的增、删、查操作很快
- Set 数据结构通过 size 来获取成员的数量，无法改变size的值。而Array数组通过 length 来获取成员的数量，length的值可以改变。

```
// Array 的操作
let arr = [];
for (let i = 0; i < 1000 * 10000; i++) {
  arr.push(i);
}

console.time("array push");
arr.push('a')
console.timeEnd("array push");

console.time("array unshift");
arr.unshift('b');
console.timeEnd("array unshift");

// Set 的操作
let s = new Set();
for (let i = 0; i < 1000 * 10000; i++) {
  s.add(i);
}

console.time("set add");
s.add('a');
console.timeEnd("set add");

// array push: 0.0078125 ms
// array unshift: 10.2490234375 ms
// set add: 0.008056640625 ms
```

数组的 `push` 方法和 Set 的 `add` 方法，时间复杂度是一样的数量级，而数组的 `unshift` 方法时间复杂度比前者大了很多。因为 set 是无序的，添加成员很快，数组的 `push` 是在最后追加成员操作也很快，但是数组的 `unshift` 的是在数组前面加上一个成员，这导致了后面全部成员的位置都要移动操作成本是很高了，使得时间复杂度增加了。

```
// Array 的操作
let arr = [];
for (let i = 0; i < 1000 * 10000; i++) {
  arr.push(i);
}
```

```
}

console.time("array includes");
arr.includes(5000000);
console.timeEnd("array includes");

// Set 的操作
let s = new Set();
for (let i = 0; i < 1000 * 10000; i++) {
  s.add(i);
}

console.time("set has");
s.has(5000000);
console.timeEnd("set has");

// array includes: 5.923095703125 ms
// set has: 0.010009765625 ms
```