

新方法的总结

一、Array.from()

- 1、作用：用于将类数组对象转化为数组；例：如下

```
let arrayLike = {
  '0': 'a',
  '1': 'b',
  '2': 'c',
  length: 3
};

let arr2 = Array.from(arrayLike);
console.log(arr2)
// ["a", "b", "c"]
```

- 2、Array.from 还可以接受第二个参数，用来对每个元素进行处理，将处理后的值放入返回的数组；例：

```
let arrayLike = {
  '0': 1,
  '1': 2,
  '2': 3,
  length: 3
};

let arr2 = Array.from(arrayLike, x => x * x);
console.log(arr2)
// [1, 4, 9]
```

- 3、Array.from()还可以将字符串转为数组，然后返回字符串的长度。例：

```
function countSymbols(string) {
  console.log(Array.from(string).length); // 6
  return Array.from(string);
}

console.log(countSymbols("string")) // ["s", "t", "r", "i", "n", "g"]
```

二、Array.of()

- 1、Array.of 方法用于将一组值，转换为数组。（总是返回参数值组成的数组。如果没有参数，就返回一个空数组）；例：

```
console.log(Array.of(3, 11, 8)) // [3,11,8]
console.log(Array.of(3)) // [3]
console.log(Array.of()) // []
```

三、fill()

- 1、fill 方法使用给定值，来填充一个数组。fill 方法还可以接受第二个和第三个参数，用于指定填充的起始位置和结束位置；例：

```
console.log(['a', 'b', 'c', 'd'].fill(3, 1, 3))  
// ["a", 3, 3, "d"]
```

四、includes()

- 1、该方法返回一个布尔值，表示某个数组是否包含给定的值。（该方法的第二个参数表示搜索的起始位置，默认为 0。如果第二个参数为负数，则表示倒数的位置，如果这时它大于数组长度（比如第二个参数为-4，但数组长度为 3），则会重置为从 0 开始。）例：

```
console.log([1, 2, 3].includes(2)); // true  
console.log([1, 2, 3].includes(4)); // false  
console.log([1, 2, 3].includes(3, 3)); // false  
console.log([1, 2, 3].includes(3, -1)); // true
```

五、entries(), keys()和 values()

- 1、entries(), keys()和 values()——用于遍历数组。它们都返回一个遍历器对象，可以用 for...of 循环进行遍历，区别是 keys()是对键名的遍历、values()是对键值的遍历，entries()是对键值对的遍历。例：

```
let arr = ['a', 'b', 'c']  
  
for (let index of arr.keys()) {  
  console.log(index);  
}  
// 0  
// 1  
// 2  
  
for (let elem of arr.values()) {  
  console.log(elem);  
}  
// 'a'  
// 'b'  
// 'c'  
  
for (let [index, elem] of arr.entries()) {  
  console.log(index, elem);  
}  
// 0 "a"  
// 1 "b"  
// 2 "c"
```

六、find()和 findIndex()

- 1、find()用于找出第一个符合条件的数组成员。它的参数是一个回调函数，所有数组成员依次执行该回调函数，直到找出第一个返回值为 true 的成员，然后返回该成员。如果没有符合条件的成员，则返回 undefined。

```
let arr = [1, 4, -5, 10, -3]
console.log(arr.find((n) => n < 0)) // -5
```

- 2、findIndex()返回第一个符合条件的数组成员的位置，如果所有成员都不符合条件，则返回-1。

```
let arr = [1, 4, -5, 10, -3]
console.log(arr.findIndex((n) => n < 0)) // 2
```

- 3、这两个方法都可以接受第二个参数，用来绑定回调函数的 this 对象。例：

```
let arr = [10, 12, 26, 15]

function f(v) {
  return v > this.age;
}

let person = { name: 'John', age: 20 };
console.log(arr.find(f, person)); // 26
console.log(arr.findIndex(f, person)); // 2
```