

12. 数组的扩展改进

学习要点：

1. 运算符扩展
2. 方法的扩展

本节课我们来开始学习 ES6 新增的数组的扩展和改进功能。

一. 运算符扩展

1. ES6 提供了 (...) 三个点将一个数组转换为逗号分割来进行处理；

```
function add(x, y) {  
    return x + y;  
}  
  
console.log(add(...[10, 20]));
```

2. 既然三点运算符通过逗号分割，那么可以想象的应用场景就随意发挥；

```
console.log(Math.max(...[1, 2, 3]));    //求最大值  
console.log([...[1, 2], ...[3, 4]]);    //合并数组
```

二. 方法的扩展

1. ES6 提供了 Array.of() 方法，它的主要目的是弥补 Array() 的不足；

```
//ES5 声明数组，当小于两位，显示长度  
let items = Array(3);  
  
//ES6 声明数组，简化版  
let items = Array.of(3, 2, 1);  
console.log(items);
```

2. ES6 提供了 Array.from() 方法，将类似数组的对象或遍历转换成真正的数组；

```
let obj = {  
    0 : 'name',  
    1 : 'age',  
    2 : 'gender',  
    length : 3,  
};  
  
let items = Array.from(obj);  
console.log(items);
```

PS：对象转换成数组要求比较严格：(1).key 必须是数值或字符串数字；
(2).length 设置长度，而且 key 在范围内；

既然要求这么严格，那什么样的场景会产生这种对象？

- (1).DOM 的 `NodeList` 集合；
- (2).ES6 新增的 `Set` 和 `Map`(后续内容)；

//返回的是类似数组的对象

```
let nl = document.querySelectorAll('p');
```

3. ES6 提供了 `find()`和 `findIndex()`方法，用于查找数组中第一个匹配的值；

```
let items = [10, 20, 30, 40, 50];  
//参数是一个回调函数，可以用箭头函数，findIndex 可以得到索引  
console.log(items.find(value => value > 19));
```

4. ES6 提供了 `fill()`方法，可以填充重写数组中的元素值；

```
let items = [1, 2, 3, 4, 5];  
items.fill('a');  
console.log(items);           //[ 'a', 'a', 'a', 'a', 'a' ]
```

```
let items = [1, 2, 3, 4, 5];  
items.fill('a', 3, 4);        //索引 3 开始，索引 4 结束  
console.log(items);           //[ 1, 2, 3, 'a', 1 ]
```

5. ES6 提供了 `copyWithin` 方法，从数组内部复制值，然后粘贴指定位置；

```
let items = [1, 2, 3, 4, 5];  
//从索引 0 开始复制值  
//然后把值从索引 2 开始粘贴  
//参数 3 设置结束粘贴索引值  
items.copyWithin(2, 0);  
console.log(items);
```