



链滴

# 你没注意到的 ES6 知识点

作者: [maixiaojie](#)

原文链接: <https://ld246.com/article/1560869506455>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



<a name="izUvE"></a>

## 增强的对象字面量

- 分配变量的简化形式

```
let x = 'a';
let y = 'b';
const obj = {
  x,
  y
}
```

- 允许在创建时定义计算成员的名称

```
const person = {
  name: 'wmai',
  surname: 'travy',
  get fullname() {
    return this.name + ' ' + this.surname;
  },
  set fullname(val) {
    let arr = val.split(' ');
    this.name = arr[0];
    this.surname = arr[1];
  }
}

console.log(person.fullname); // wmai travy
console.log(person.fullname = 'wen yujie'); // wen yujie
console.log(person);
```

通过 get 和 set 定义的 fullname 属性， 默认情况下， set 方法的返回值是 get 方法返回的相同属性的值。

<a name="w2lQ3"></a>

## Map 和 Set

可以定义哈希映射集合。

```
const pro = new Map();
pro.set('x', 'x1');
pro.set('y', 'y1');
pro.set('x1', 'x2');
pro.set('y1', 'y2');
console.log(pro.size); // 4
console.log(pro.get('x')); // x1
console.log(pro.get('a')); // undefined
console.log(pro.has('y')); // true
console.log(pro.has('a')); // false
console.log(pro.delete('x')); // true
console.log(pro.has('x')); // false

for(const item of pro) {
  console.log(item);
}
/*
[ 'y', 'y1' ]
[ 'x1', 'x2' ]
[ 'y1', 'y2' ]
*/
```

Map 提供了 set, get, has, delete 等方法，还有 size 属性。可以用 for...of 来遍历。

Map 可以使用函数和对象作为 key，这是普通对象不具有的。可以利用此特性来创建一个微测试框架。

```
const test = new Map()
test.set( () => 2+2, 4);
test.set( () => 2*2, 4);
test.set( () => 2/2, 2);

for(const entry of test) {
  console.log(entry[0]() === entry[1] ? 'PASS' : 'FAIL');
}
/*
PASS
PASS
FAIL
*/
```

Set 可以构建一个集合，所有的 key 是唯一的。key 允许是对象或者函数。

```
const s = new Set([0,1, 3, 4]);
```

```
console.log(s.add(1)); // will not be added
console.log(s.has(2)); // false
console.log(s.delete(3)); // true
for(const entry of s) {
  console.log(entry); // 0 1 4
}

<a name="cOPC2"></a>
```

## WeakMap 和 WeakSet

```
const set = new Set([0, 1, 2, 3]);
let obj = {};
const map = new WeakMap();
map.set(obj, {key: 'some value'});
console.log(map.get(obj)); // { key: 'some value' }
obj = undefined;
console.log(map.get(obj)); // undefined
```

WeakMap 没有办法迭代所有条目，key 只能为对象。

```
let obj1 = {key: 'val1'};
let obj2 = {key: 'val2'};
const ss = new WeakSet([obj1, obj2]);
console.log(ss.has(obj1)); // true
obj1 = undefined;
console.log(ss.has(obj1)); // false
```

WeakSet 只允许存储对象且不能重复，允许内部对象在只剩下引用的时候被垃圾回收。

```
<a name="Z7ZCh"></a>
```

## 其他特性

- 模版字面量
- Promise
- 默认方法参数
- 剩余参数
- 扩展运算符
- 解构
- new.target
- Proxy
- Reflect
- Symbol