Js篇

1.冒泡：

2.捕获：

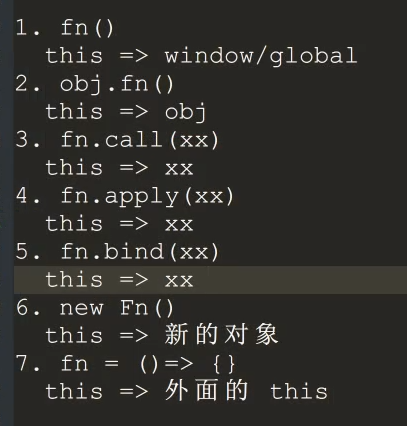
1. 深浅拷贝

浅拷贝：1. Object.assign

深拷贝：1.插件使用 lodash 的深拷贝函数cloneDeep。

2.对象的深拷贝可以通过 JSON 对象的两个方法，一是将 js 对象转成字符串对象的 JSON.stringify 方法，另一个是将 js 字符串对象转换成 js 对象的 JSON.parse 方法。

1. Promise
2. **解构赋值**
3. **增强对象的字面量**
4. **箭头函数（this指向问题）**



1. 块作用域变量声明\*\***let**\*\*和 \*\***const**\*\***var**
2. **判断数据类型**

typeof 判断简易数据类型，可以判断function ，但不能判断复杂数据类型

instanceof 因为A instanceof B 可以判断A是不是B的实例，返回一个布尔值， 由构造类型判断出数据类型

根据对象的contructor判断

通过Object下的toString.call()方法来判断

1. .**事件队列事件循环机制**
2. 原型原型链
3. 闭包
4. call apply bind区别和实现原理
5. 数据类型
6. **数组去重**

1、使用ES6中的set方法

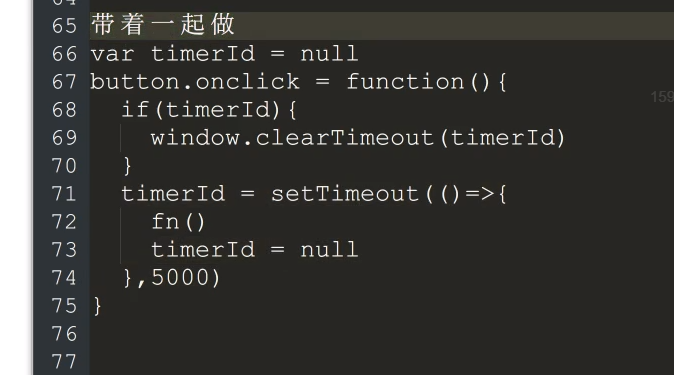
2、indexOf()方法

3、利用数组的sort()方法（相邻元素比对法）

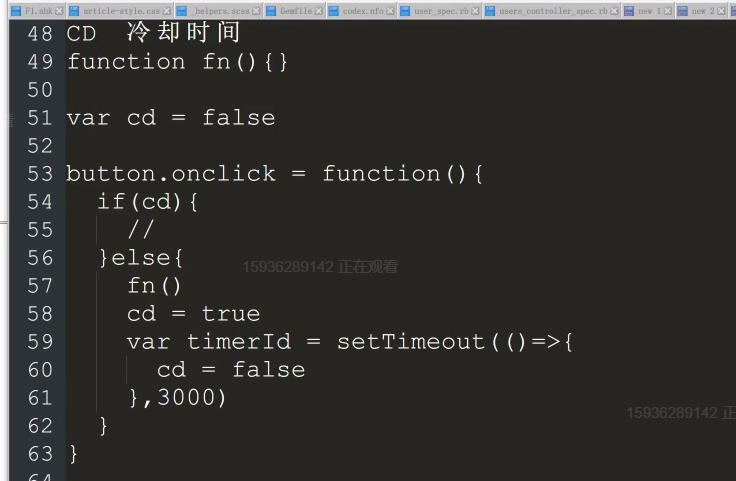
4、利用数组的includes

5、利用双重for循环、再用splice方法去重（ES5常用）、函数递归

1. **防抖**



1. **节流**



1. **es6新特性**
2. 判断是不是数组
3. **js的垃圾回收机制是什么原理**

1、全局变量引起的内存泄露

2、闭包引起的内存泄露：慎用闭包

3、dom清空或删除时，事件未清除导致的内存泄漏

3、循环引用带来的内存泄露