**v8内存管理**

1. 内存的历程
2. V8内存的优点
3. 解析v8的内存管理
4. V8内存管理算法  
    <https://www.yuque.com/u268531/kb/dkuz88>  
     
   [https://www.yuque.com/lialvin/notes/ihk5xf#t15](https://www.yuque.com/lialvin/notes/ihk5xf" \l "t15)

Js元编程和proxy  
1.什么是元编程，这非常类似Java中的动态代理模式，

**元编程是编写在运行时操纵语言构件的代码**

Proxy 是什么：Proxy 用于修改某些操作的默认行为，等同于在语言层面做出修改，所以属于一种“元编程”（meta programming），即对编程语言进行编程。

[Symbol](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Symbol" \t "https://segmentfault.com/a/_blank)、[Reflect](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/reflect" \t "https://segmentfault.com/a/_blank) 和 [Proxy](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Proxy" \t "https://segmentfault.com/a/_blank) 是属于 ES6 元编程范畴的，能“介入”的对象底层操作进行的过程中，并加以影响。元编程中的 ****元**** 的概念可以理解为 ****程序**** 本身。

一般代码的操作对象是数据  
元编程操作的对象是其他代码

"元编程"比"我们手写代码"多提供了一个抽象层次

### **Proxy 原语**

**new** Proxy("To be, or not to be...", {});​*// TypeError: Cannot create proxy with a non-object as target or handler*

1. 有什么用，可以干什么

let api = { \_apiKey: '123abc456def', getUsers: function() { /\* ... \*/ }, getUser: function(userId) { /\* ... \*/ }, setUser: function(userId, config) { /\* ... \*/ } }; function logMethodAsync(timestamp, method) { setTimeout(function() { console.log(`${timestamp} - Logging ${method} request asynchronously.`); }, 0) } api = new Proxy(api, { get: function(target, key, proxy) { var value = target[key]; return function(...arguments) { logMethodAsync(new Date(), key); return Reflect.apply(value, target, arguments); }; } }); api.getUsers();著作权归作者所有。  
商业转载请联系作者获得授权,非商业转载请注明出处。  
原文: <https://www.w3cplus.com/javascript/use-cases-for-es6-proxies.html> © [w3cplus.com](https://www.w3cplus.com/)

1. 怎么用
2. 有哪些属性
3. vue3中已经使用了Proxy ，解决了什么问题

虽然Object.defineProperty通过为属性设置getter/setter能够完成数据的响应式，但是它并不算是实现数据的响应式的完美方案，某些情况下需要对其进行修补或者hack，这也是它的缺陷，主要表现在两个方面：

1. **无法检测到对象属性的新增或删除**

**vue.set();**

data.location = { x: 100, y: 100 } data.location = {...data, z: 100} 重新赋值

1. **不能监听数组的变化**

**解决方法：**push/pop/shift/unshift/splice/sort/reverse

1. js和元编程 元编程在 ES6 体现最为突出的是 Proxy 的应用

元编程的小小实践

1. over  
   参考网址：
2. <http://caibaojian.com/es6/proxy.html>  
   [https://segmentfault.com/a/1190000016133613#item-5](https://segmentfault.com/a/1190000016133613" \l "item-5)  
   <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Meta_programming>

返回的是一个新的对象 是个对象就行，能点就行

DeleteProperty(){}删除

Has() //有咩有这个东西

Apply() //调用函数的处理

Target propterty

Vue 不存在的返回的报错

Console.warn(`${property}不在此对象上`)

console.log(“^\_^”）

Function（attr=>{},...children){

Document.createElement(proptery);

For(let key of Object.keys(attr)){

El.setAttribute(key,attr[key]);

}

}

Document.body.appendChild(oDIv);

Window.onload(){

Document.body.appendChild(oDIv);

}