



链滴

如何阻止修改一个 JavaScript 对象

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1595652158216>

来源网站: [链滴](#)

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

2020-07-24

描述

在 JavaScript 中如何真正阻止对一个对象的修改。

const 也可修改

说明

JavaScript 中的对象是可修改的，无论你是否将他们定义为 `const` 变量。事实上，当使用 `const` 定一个对象时，只能阻止他再次被赋值。然而，你可以给一个 `const` 对象或数组的属性进行再次赋值。

代码

```
const myObj = { a: 10, b: 20, c: 30 };  
myObj.a = 12; // {a: 12, b: 20, c: 30}
```

```
const myArr = [15, 25, 35];  
myArr[1] = 28; // [15, 28, 35]
```

Object.freeze 依旧还能修改

说明

为了让对象不可修改，我们可以使用 `Object.freeze()`，他可以在某种程度上阻止添加新的属，阻止删及修改已存在的属性。虽然 `Object.freeze()` 能为某些情况提供解决方案，但是到嵌套的下一层中，个解决方案就会存在问题。因为他实际上只能执行浅冻结。这也就意味着对象或数组的属性依然可以修改。

代码

```
const myObj = {  
  a: 1,  
  b: 'hello',  
  c: [0, 1, 2],  
  d: { e: 1, f: 2 }  
};  
Object.freeze(myObj);
```

```
myObj.a = 10;  
myObj.b = 'hi';  
myObj.c[1] = 4;  
myObj.d.e = 0;
```

```
/*  
myObj = {  
  a: 1,  
  b: "hello",
```

```
    c: [0, 4, 2],
    d: { e: 0, f: 2 }
  }
  */
```

彻底阻止修改

说明

正如你所见，`Object.freeze()` 起到了一定的作用，但他仅只能对一个对象进行浅冻结。为了解决这个问题，我们可以使用递归来检查每一个属性自身是否为一个对象。如果该对象的 `Object.isFrozen()` 返回 `alse`，我们就对他使用 `Object.freeze()`

代码

```
const myObj = {
  a: 1,
  b: 'hello',
  c: [0, 1, 2],
  d: { e: 1, f: 2 }
};

const deepFreeze = obj => {
  Object.keys(obj).forEach(prop => {
    if (typeof obj[prop] !== 'object' && !Object.isFrozen(obj[prop])){
      deepFreeze(obj[prop]);
    }
  });
  return Object.freeze(obj);
};

deepFreeze(myObj);

myObj.a = 10;
myObj.b = 'hi';
myObj.c[1] = 4;
myObj.d.e = 0;

/*
myObj = {
  a: 1,
  b: "hello",
  c: [0, 1, 2],
  d: { e: 1, f: 2 }
}
*/
```

在上面的示例中，我们使用了上一章所描述的技术来确保给定的对象被深度冻结。你可以在 [深度冻结](#) 中查看完整的代码和更多示例片段。

返回总目录

