

Object.preventExtensions()

概述

Object.preventExtensions() 方法让一个对象变的不可扩展，也就是永远不能再添加新的属性。

语法

```
Object.preventExtensions(obj)
```

参数

obj

将要变得不可扩展的对象

描述

如果一个对象可以添加新的属性，则这个对象是可扩展的。**preventExtensions** 可以让这个对象变的不可扩展，也就是不能再有新的属性。需要注意的是不可扩展的对象的属性通常仍然可以被删除。尝试给一个不可扩展对象添加新属性的操作将会失败，不过可能是静默失败，也可能会抛出 **TypeError** 异常（**严格模式**）。

Object.preventExtensions 只能阻止一个对象不能再添加新的自身属性，仍然可以为该对象的原型添加属性。然而**Object.preventExtensions**会阻止一个对象将__proto__属性重新指向另一个对象。

在 ECMAScript 5 中可扩展的对象可以变得不可扩展，但反过来不行。

例子

```
1 // Object.preventExtensions将原对象变的不可扩展,并且返回原对象.
2 var obj = {};
3 var obj2 = Object.preventExtensions(obj);
4 assert(obj === obj2);
5
6 // 字面量方式定义的对象默认是可扩展的.
7 var empty = {};
```

2017/10/13Object.preventExtensions() - JavaScript | MDN

```
8  assert(Object.isExtensible(empty) === true);
9
10 // ...但可以改变.
11 Object.preventExtensions(empty);
12 assert(Object.isExtensible(empty) === false);
13
14 // 使用Object.defineProperty方法为一个不可扩展的对象添加新属性会抛出异常.
15 var nonExtensible = { removable: true };
16 Object.preventExtensions(nonExtensible);
17 Object.defineProperty(nonExtensible, "new", { value: 8675309 }); // 抛出T
18
19 // 在严格模式中,为一个不可扩展对象的新属性赋值会抛出TypeError异常.
20 function fail()
21 {
22     "use strict";
23     nonExtensible.newProperty = "FAIL"; // throws a TypeError
24 }
25 fail();
26
27 // 一个不可扩展对象的原型是不可更改的,__proto__是个非标准魔法属性,可以更改一个对
28 var fixed = Object.preventExtensions({});
29 fixed.__proto__ = { oh: "hai" }; // 抛出TypeError异常
```

规范

Specification	Status	Comment
ECMAScript 5.1 (ECMA-262) Object.preventExtensions	<div> </div> ST Standard	Initial definition. Implemented in JavaScript 1.8.5
ECMAScript 2015 (6th Edition, ECMA-262) Object.preventExtensions	<div> </div> ST Standard	

浏览器兼容性

Desktop

Mobile

Feature	Firefox (Gecko)	Chrome	Internet Explorer	Opera	Safari
Basic support	4 (2.0)	6	9	未实现	5.1

相关链接

- `Object.isExtensible`
- `Object.isSealed`
- `Object.isFrozen`
- `Object.seal`
- `Object.freeze`