Object.prototype.constructor

返回创建实例对象的 `Object` 构造函数的引用。注意，此属性的值是对函数本身的引用，而不是一个包含函数名称的字符串。该值为只读的原始类型，如 `1` ， `true` 和 `"test"` 。

描述

所有对象都会从它的原型上继承一个 `constructor` 属性：

```
var o = {};
o.constructor === Object; // true

var o = new Object;
o.constructor === Object; // true

var a = [];
a.constructor === Array; // true

var a = new Array;
a.constructor === Array // true

var n = new Number(3);
n.constructor === Number; // true
```

示例

打印一个对象的构造函数

以下示例创建一个原型， `Tree` ，以及该类型的对象，即 `theTree` 。 然后打印 `theTree` 对象的 `constructor` 属性。

```
function Tree(name) {
   this.name = name;
}

```

```
var theTree = new Tree("Redwood");
console.log( "theTree.constructor is " + theTree.constructor );
```

打印输出：

```
theTree.constructor is function Tree(name) {
    this.name = name;
}
```

改变对象的 constructor

下面的例子展示了如何修改基本类型对象的 constructor 属性的值。只有 true, 1 和 "test" 的不受影响，因为创建他们的是只读的原生构造函数（native constructors）。这个例子也说明了依赖一个对象的 constructor 属性并不安全。

以下示例显示如何修改泛型对象的构造函数值。 只有 `true` , `1` 和 `"test"` 不受影响，因为他们有只读的原生构造函数。 这个例子也表明依靠对象的 `constructor` 属性并不总是安全的。

```
function Type() { };

var types = [
    new Array,
    [],
    new Boolean,
    true,        // remains unchanged
    new Date,
    new Error,
    new Function,
    function(){},
    Math,
    new Number,
    1,           // remains unchanged
    new Object,
    {},
    new RegExp,
    /(?:)/,
    new String,
    "test"       // remains unchanged
];

for(var i = 0; i < types.length; i++) {
```

```
    types[i].constructor = Type;
    types[i] = [ types[i].constructor, types[i] instanceof Type, types[i]
};

console.log( types.join("\n") );
```

此示例显示以下输出：

```
function Type() {},false,
function Type() {},false,
function Type() {},false,false
function Boolean() {
    [native code]
},false,true
function Type() {},false,Mon Sep 01 2014 16:03:49 GMT+0600
function Type() {},false,Error
function Type() {},false,function anonymous() {

}
function Type() {},false,function () {}
function Type() {},false,[object Math]
function Type() {},false,0
function Number() {
    [native code]
},false,1
function Type() {},false,[object Object]
function Type() {},false,[object Object]
function Type() {},false,/(?:)/
function Type() {},false,/(?:)/
function Type() {},false,
function String() {
    [native code]
},false,test
```

规范

Specification	Status	Comment
ECMAScript 1st Edition (ECMA-262)	ST Standard	Initial definition. Implemented in JavaScript 1.1.

ECMAScript 5.1 (ECMA-262) Object.prototype.constructor	ST Standard	
ECMAScript 2015 (6th Edition, ECMA-262) Object.prototype.constructor	ST Standard	
ECMAScript Latest Draft (ECMA-262) Object.prototype.constructor	LS Living Standard	

浏览器兼容

Desktop Mobile

Feature	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari
Basic Support	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)