

Object.prototype.hasOwnProperty()

hasOwnProperty() 方法会返回一个布尔值，指示对象是否具有指定的属性作为自身（不继承）属性。

语法

```
obj.hasOwnProperty(prop)
```

参数

prop

要检测的属性 [字符串](#) 名称或者 [Symbol](#) 。

返回值

用来判断某个对象是否含有指定的属性的 [Boolean](#) 。

描述

所有继承了 [Object](#) 的对象都会继承到 **hasOwnProperty** 方法。这个方法可以用来检测一个对象是否含有特定的自身属性；和 [in](#) 运算符不同，该方法会忽略掉那些从原型链上继承到的属性。

示例

使用 hasOwnProperty 方法判断属性是否存在

下面的例子检测了对象 `o` 是否含有自身属性 `prop`：

```
1 | o = new Object();
2 | o.prop = 'exists';
3 |
4 | function changeO() {
5 |     o.newprop = o.prop;
6 |     delete o.prop;
```

```
7   }  
8  
9   o.hasOwnProperty('prop');    // 返回 true  
10  change0();  
11  o.hasOwnProperty('prop');    // 返回 false
```

自身属性与继承属性

下面的例子演示了 `hasOwnProperty` 方法对待自身属性和继承属性的区别：

```
1  o = new Object();  
2  o.prop = 'exists';  
3  o.hasOwnProperty('prop');      // 返回 true  
4  o.hasOwnProperty('toString');  // 返回 false  
5  o.hasOwnProperty('hasOwnProperty'); // 返回 false
```

遍历一个对象的所有自身属性

下面的例子演示了如何在遍历一个对象的所有属性时忽略掉继承属性，注意这里 `for...in` 循环只会遍历可枚举属性，所以不应该基于这个循环中没有不可枚举的属性而得出 `hasOwnProperty` 是严格限制于可枚举项目的（如同 `Object.getPrototypeOfNames()`）。

```
1  var buz = {  
2    fog: 'stack'  
3  };  
4  
5  for (var name in buz) {  
6    if (buz.hasOwnProperty(name)) {  
7      alert("this is fog (" + name + ") for sure. Value: " + buz[name]);  
8    }  
9    else {  
10     alert(name); // toString or something else  
11    }  
12  }
```

使用 `hasOwnProperty` 作为属性名

JavaScript 并没有保护 `hasOwnProperty` 属性名，因此某个对象是有可能存在使用这个属性名的属性，使用外部的 `hasOwnProperty` 获得正确的结果是需要：

```
1  var foo = {
2      hasOwnProperty: function() {
3          return false;
4      },
5      bar: 'Here be dragons'
6  };
7
8  foo.hasOwnProperty('bar'); // 始终返回 false
9
10 // 如果担心这种情况，可以直接使用原型链上真正的 hasOwnProperty 方法
11 ({}).hasOwnProperty.call(foo, 'bar'); // true
12
13 // 也可以使用 Object 原型上的 hasOwnProperty 属性
14 Object.prototype.hasOwnProperty.call(foo, 'bar'); // true
```

规范

Specification	Status	Comment
ECMAScript 3rd Edition (ECMA-262)	<div><div></div>STStandard</div>	Initial definition. Implemented in JavaScript 1.5.
ECMAScript 5.1 (ECMA-262) Object.prototype.hasOwnProperty	<div><div></div>STStandard</div>	
ECMAScript 2015 (6th Edition, ECMA-262) Object.prototype.hasOwnProperty	<div><div></div>STStandard</div>	
ECMAScript Latest Draft (ECMA-262) Object.prototype.hasOwnProperty	<div><div></div>LSLiving Standard</div>	

浏览器兼容性

	Desktop	Mobile			
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)

相关链接

- 属性的可枚举性和所有权
- `Object.getOwnPropertyNames()`
- `for...in`
- `in`
- JavaScript 教程: 再谈继承