Object.prototype.propertyIsEnumerable()

概述

`propertyIsEnumerable()` 方法返回一个布尔值，表明指定的属性名是否是当前对象可枚举的自身属性。

语法

```
obj.propertyIsEnumerable(prop)
```

参数

`prop`
需要测试的属性名。

返回值

一个 **`Boolean`** 用来表明指定的属性名是否可枚举

描述

每个对象都有 `propertyIsEnumerable` 方法。该方法可以判断出指定对象里的属性是否可枚举，也就是说该属性是否可以通过 `for...in` 循环等遍历到，不过有些属性虽然可以通过 `for...in` 循环遍历到，但因为它们不是自身属性，而是从原型链上继承的属性,所以该方法也会返回 `false` 。如果对象没有指定的属性，该方法返回 `false` 。

例子

`propertyIsEnumerable` 方法的基本用法

下面的例子演示了 `propertyIsEnumerable` 方法在普通对象和数组上的基本用法:

```
var o = {};
var a = [];
o.prop = 'is enumerable';
a[0] = 'is enumerable';

o.propertyIsEnumerable('prop');   //  返回 true
a.propertyIsEnumerable(0);        // 返回 true
```

用户自定义对象和引擎内置对象

下面的例子演示了用户自定义对象和引擎内置对象上属性可枚举性的区别.

```
var a = ['is enumerable'];

a.propertyIsEnumerable(0);          // 返回 true
a.propertyIsEnumerable('length');   // 返回 false

Math.propertyIsEnumerable('random');   // 返回 false
this.propertyIsEnumerable('Math');     // 返回 false
```

自身属性和继承属性

```
var a = [];
a.propertyIsEnumerable('constructor');         // 返回 false

function firstConstructor() {
  this.property = 'is not enumerable';
}

firstConstructor.prototype.firstMethod = function() {};

function secondConstructor() {
  this.method = function method() { return 'is enumerable'; };
}

secondConstructor.prototype = new firstConstructor;
secondConstructor.prototype.constructor = secondConstructor;

var o = new secondConstructor();
o.arbitraryProperty = 'is enumerable';

o.propertyIsEnumerable('arbitraryProperty');   // 返回 true
o.propertyIsEnumerable('method');              // 返回 true
o.propertyIsEnumerable('property');            // 返回 false

o.property = 'is enumerable';

o.propertyIsEnumerable('property');            // 返回 true
```

```
// 这些返回fasle，是因为，在原型链上propertyIsEnumerable不被考虑
// (尽管最后两个在for-in循环中可以被循环出来)。
o.propertyIsEnumerable('prototype');   // 返回 false (根据 JS 1.8.1/FF3.6)
o.propertyIsEnumerable('constructor'); // 返回 false
o.propertyIsEnumerable('firstMethod'); // 返回 false
```

规范

Specification	Status	Comment
ECMAScript 3rd Edition (ECMA-262)	ST Standard	Initial definition.
ECMAScript 5.1 (ECMA-262) Object.prototype.propertyIsEnumerable	ST Standard	
ECMAScript 2015 (6th Edition, ECMA-262) Object.prototype.propertyIsEnumerable	ST Standard	
ECMAScript Latest Draft (ECMA-262) Object.prototype.propertyIsEnumerable	LS Living Standard	

浏览器兼容性

Desktop Mobile

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

Gecko-specific notes

- 从 JavaScript 1.8.1 (in Firefox 3.6) 开始， `propertyIsEnumerable("prototype")` 返回 false，不再是 true；这与 ECMAScript 5 的结果一致。

相关链接

- Enumerability and ownership of properties
- for...in
- `Object.keys()`
- `Object.defineProperty()`