



# 深入Object.defineProperty



jCodeLife



0.342

2020.09.17 17:32:57

字数 1,387

阅读 38

[编辑文章](#)

从名字就可以看出，`Object.defineProperty()` 其实就是用于定义一个对象的属性的，有三个点

1. 给哪个对象添加或修改属性
2. 添加或者修改的属性名
3. 属性名对应的 `value` 值等相关配置

三个点也是对应3个必填参数

```
1 let obj = {}
2 Object.defineProperty(obj, 'name', {
3   value: 'alice'
4 });
5 console.log(obj.name)//alice
```

上述代码中，表示往 `obj` 对象中添加一个属性 `name`，值为 `alice`

`Object.defineProperty()` 三个传参都必填，如果参数漏填或者格式不正确，例如我第三个参数没写，就会直接报错 `Uncaught TypeError: Property description must be an object: undefined` 但平时我们往往使用 `.操作符` 来给一个对象添加属性或者修改已有属性的属性值

```
1 let obj = {}
2 obj.name = 'alice'
3 console.log(obj.name)//alice
```

上述代码跟前面的代码实现的功能是一致的，都是给 `obj` 对象添加一个属性 `name`，值为 `alice` 到这里你可能会由一个疑问，居然平时使用 `.` 的形式可以代替 `Object.defineProperty()` 的功能，那是不是没有必要使用 `Object.defineProperty()`？

我们要知道，每个语法都有自己的使用场景的！其实如果仅仅是给对象的一个属性对应的 `value` 值，确实没有必要用到 `Object.defineProperty()`，直接使用 `对象.属性 = xxx` 的形式就好了。但是如果要给属性配置一些特性，那就得用它了，因为 `Object.defineProperty()` 的第三个参数就是属性对应的一些配置，也被称为属性的描述 `descriptor`，其中包含例子中写的属性值 `value`

值得注意的是：默认情况下，使用 `Object.defineProperty()` 添加的属性值是不可修改（`immutable`）的

如果再通过 `Object.defineProperty()` 重新修改 `Object.defineProperty()` 定义过的属性，会被提示不能重复定义：`Uncaught TypeError: Cannot redefine property: xxx`

如果通过 `对象.属性 = xxx` 的形式尝试修改用 `Object.defineProperty()` 新增的属性，虽然没有语法错误，但是当再次访问时，结果还是 `Object.defineProperty()` 之前定义的值

```
1 let obj = {}
2 Object.defineProperty(obj, 'name', {
3   value: 'alice'
4 });
5 obj.name = 'tom'
6 console.log(obj.name)//alice
```

上述代码中，通过 `obj.name='tom'` 再次修改对应的值，访问时还是 `Object.defineProperty()` 之前定义的值 `alice`；

## 推荐阅读

[5个技巧让你更好的编写](#)[JavaScript\(ES6\) 中条件语句](#)[阅读 92](#)[Es6基础语法](#)[阅读 1,172](#)[js 判断字符串中是否包含某个字符串](#)[阅读 370](#)[vue项目实现登录携带token](#)[阅读 347](#)[JavaScript基础篇（三）作用域和闭包](#)[阅读 57](#)



的

```
1 let obj = {}
2 obj.name = 'lbj'
3 console.log(obj.name)//lbj
4 Object.defineProperty(obj, 'name', {
5   value: 'alice'
6 });
7 console.log(obj.name)//alice
8 obj.name = 'tom'
9 console.log(obj.name)//tom
10 Object.defineProperty(obj, 'name', {
11   value: 'aaa'
12 });
13 console.log(obj.name)//aaa
```

以上代码中，由于最开始就存在这个属性，所以可以通过 `obj.name=xxx` 或者用 `Object.defineProperty()` 重复修改

下面回过头来看，属性描述中还可以配置哪一些？

对象里目前存在的属性描述符有两种主要形式：`数据描述符` 和 `存取描述符`。数据描述符是一个具有值的属性，该值可以是可写的，也可以是不可写的。存取描述符是由 `getter` 函数和 `setter` 函数所描述的属性。一个描述符只能是这两者其中之一；不能同时是两者。

这两种描述符都是对象。它们共享以下可选键值

#### 1. `configurable`

表示可配置的，默认值为 `false`；当值为 `true` 时，该属性的描述符才能够被改变，同时该属性也能从对应的对象中删除

#### 2. `enumerable`

表示可枚举的，默认为 `false`；当设置为 `true` 时，该属性才会在对象枚举时枚举到

数据描述符还具有可选键值：

#### 1. `value`

表示属性对应的值，默认 `undefined`

#### 2. `writable`

表示可写的，当 `writable` 为 `true` 时，属性值才能被赋值运算符更改成功；默认值为 `false`，所以默认情况下，使用 `Object.defineProperty()` 添加的属性值是不可修改（`immutable`）的

存取描述符还具有以下可选键值：

#### 1. `get`

当获取该属性时，会执行 `get` 函数，`get` 返回值就是属性值

```
1 let obj = {}
2 Object.defineProperty(obj, 'name', {
3   configurable: true,
4   enumerable: true,
5   get: function () {
6     console.log(`你访问了obj.name属性哦`);
7     return 'tom';
8   }
9 })
10 console.log(obj.name)
```

### 推荐阅读

5个技巧让你更好的编写

JavaScript(ES6) 中条件语句

阅读 92

Es6基础语法

阅读 1,172

js 判断字符串中是否包含某个字符串

阅读 370

vue项目实现登录携带token

阅读 347

JavaScript基础篇（三）作用域和闭包

阅读 57



如果 `get` 没有返回值，默认属性值为 `undefined`

`get` 执行时不用传入任何参数，需要注意的是：里头的 `this` 指向一般是该对象，但不一定是！

## 2. `set`

当修改对象的属性值时会调用 `set` 函数

```
1 let obj = {}
2 Object.defineProperty(obj, 'name', {
3   configurable: true,
4   enumerable: true,
5   set: function (val) {
6     console.log(`设置obj.name的值为: ${val}`)
7   }
8 })
9 obj.name = 'aaa'//
```

上面代码中，设置 `obj.name` 的值为 `aaa`，默认会将设置的值 `aaa` 作为 `set` 函数的第一个参数传递进去

小结一下：

### 1. 数据描述符可以拥有：

`configurable`、`enumerable`、`value`、`writable`

存取描述符可以拥有：

`configurable`、`enumerable`、`get`、`set`

### 2. `configurable`、`enumerable` 和 `writable` 的默认值都是 `false`；

`value`、`get` 和 `set` 的默认值为 `undefined`

### 3. 如果一个描述符不具有 `value`、`writable`、`get` 和 `set` 中的任意一个键，那么它将被认为是一个数据描述符。如果一个描述符同时拥有 `value` 或 `writable` 和 `get` 或 `set` 键，则会报错

### 4. `Object.defineProperty()` 的返回值就是属性要添加在的对象，其实就是第一个参数



3人点赞 >



笔记 (2)



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife 书山有路勤为径，学海无涯苦作舟

总资产73 (约5.46元) 共写了13.6W字 获得340个赞 共29个粉丝

## 推荐阅读

5个技巧让你更好的编写

JavaScript(ES6) 中条件语句

阅读 92

Es6基础语法

阅读 1,172

js 判断字符串中是否包含某个字符串

阅读 370

vue项目实现登录携带token

阅读 347

JavaScript基础篇（三）作用域和闭包

阅读 57



写下你的评论...



写下你的评论...

评论 0

赞 3






被以下专题收入，发现更多相似内容

投稿管理

+ 收入我的专题

 前段开发那些事儿

## 推荐阅读

5个技巧让你更好的编写

JavaScript(ES6) 中条件语句

阅读 92

Es6基础语法

阅读 1,172

js 判断字符串中是否包含某个字符串

阅读 370

vue项目实现登录携带token

阅读 347

JavaScript基础篇（三）作用域和闭包

阅读 57

写下你的评论...

 评论0

 赞3

...