博学谷
www.boxuegu.com

第八章 Symbol和Symbol属性

一样的在线教育，不一样的教学品质

目录 Contents

- Symbol简介与基本使用
- Symbol应用场景（实战应用）
- Symbol属性名遍历

Symbol简介与基本使用

Symbol是一种**数据类型**，是JavaScript语言的**第7种数据类型**；前6种分别是:undefined，null，布尔值，字符串，数值和对象。

特征：

Symbol类型的值**通过Symbol函数生成**，其值是独一无二的，**具有唯一性**；可以保证对象中属性名称的唯一。

Symbol简介与基本使用

创建Symbol类型的变量，并打印输出：

```
let s = Symbol();
let s1 = Symbol();
console.log(s);
console.log(s1);
```

输出结果都是：Symbol( )。

优化：

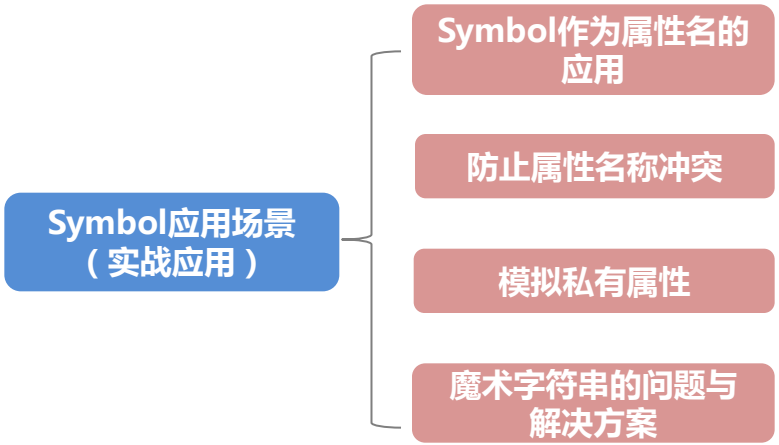
Symbol( )函数**接受一个字符串作为参数**，该参数表示对Symbol的描述；主要是为了在控制台进行输出打印时，能够**区分Symbol最终属于哪个变量**。

```
let s = Symbol('s');
let s1 = Symbol('s1');
console.log(s);
console.log(s1);
```

目录 Contents

- Symbol简介与基本使用
- Symbol应用场景（实战应用）
- Symbol属性名遍历

Symbol应用场景


Symbol应用场景
（实战应用）
Symbol作为属性名的应用
防止属性名称冲突
模拟私有属性
魔术字符串的问题与解决方案

1.Symbol作为属性名的应用

由于Symbol的值是唯一的，并且能够保证对象中不会出现同名的属性。

添加属性的三种方式：

1）通过方括号的方式给对象添加属性；

2）创建对象时直接添加属性；

3）通过Object.defineProperty方法添加属性；

2.防止属性名称冲突

示例：

定义一个对象，动态的向对象中添加一个id属性：

```
let obj = {
            name: 'zs',
            age: 18
        }
        function test1(obj) {
            obj.id = 42;
        }
        function test2(obj) {
            obj.id = 369;
        }
        test1(obj);
        test2(obj);
        console.log(obj);
```

问题：由于test2( )函数后执行，会将test1( )函数创建的id属性的值覆盖掉。

2.防止属性名称冲突

解决方案：使用Symbol作为属性名。

```
let obj = {
            name: 'zs',
            age: 18
        }
        let mySymbol = Symbol('lib1');

        function test1(obj) {
            obj[mySymbol] = 42;

        }
        let mySymbol2 = Symbol('lib2');
        function test2(obj) {
            obj[mySymbol2] = 369;

        }
        test1(obj);
        test2(obj);
        console.log(obj);
```

3.模拟私有属性

所谓**私有属性**，表示**只在当前的模块文件中可用，在其它文件中不可用。**

1）先定义一个UserInfo.js文件，代码如下：

```
let userName = Symbol('userName');
export class UserInfo {
    [userName]() {
        return 'zhangsan';
    }
    show() {
        console.log(this[userName]());
    }
}
```

3.模拟私有属性

2）调用模块：

```
<script type="module">
    let userName = Symbol('userName');
     import { UserInfo } from './UserInfo.js';
     let user = new UserInfo();
     user.show();
     console.log(user[userName])
</script>
```

注意1：script标签中要**加 type=“module”表示使用模块；**

注意2：以上代码是在微软Edge浏览器 15版本以上进行测试；

在地址栏中输入 “about:flags ” ，在打开的窗口中启用“实验性 JavaScript 功能” ，Edge浏览器才支持ES6模块编程。

4.魔术字符串的问题与解决方案

所谓魔术字符串，指**在代码中多次出现，与代码形成强耦合的某一个字符串或者是数值。**

```
function getArea(t, options) {
        let area = 0;
        switch (t) {
            case "add": // 魔术字符串
                area = 5 + options.width + options.height;
                break;
        }
        return area;
    }

    getArea("add", {
        width: 100,
        height: 100
    }); // 魔术字符串
```

目录 Contents

- Symbol简介与基本使用
- Symbol应用场景
- Symbol属性名遍历

属性名的遍历

Symbol作为属性名后，**不能使用for...in ，for...of，Object.keys( ) 来遍历属性**。

获取对象中的 Symbol 属性：使用 **Object.getOwnPropertySymbols( )方法**。

示例：

```
let mySymbol = Symbol();
        let obj = {
            userName: 'zs',
            age: 18,
            [mySymbol]: 'hello'

        }
        console.log(Object.getOwnPropertySymbols(obj));
```

一样的在线教育，不一样的教学品质