

后盾人 人人做后盾

www.houdunren.com

ES6语法

后盾人 2011-2018

- null
- undefined
- 布尔值 (Boolean)
- 数组 (Array)
- 字符串 (String)
- 对象 (Object)
- Symbol

ES6数据类型

- ES5 的对象属性名都是字符串，这容易造成属性名的冲突。比如，你使用了一个他人提供的对象，但又想为这个对象添加新的方法（mixin 模式），新方法的名字就有可能与现有方法产生冲突。如果有一种机制，保证每个属性的名字都是独一无二的就好了，这样就从根本上防止属性名的冲突。这就是 ES6 引入 Symbol 的原因。
- ES6 引入了一种新的原始数据类型 Symbol，表示独一无二的值
- 它是一种类似于字符串的数据类型

Symbol 类型

- Symbol函数可以接受一个字符串作为参数，表示对 Symbol 实例的描述，主要是为了在控制台显示，或者转为字符串时，比较容易区分。
- Symbol类型的值独一无二，不与任何其他值相等
- Symbol 值不能与其他类型的值进行运算。

Symbol类型

- 由于每一个 Symbol 值都是不相等的，这意味着 Symbol 值可以作为标识符，用于对象的属性名，就能保证不会出现同名的属性。这对于一个对象由多个模块构成的情况非常有用，能防止某一个键被不小心改写或覆盖。

作为属性名的Symbol

```
let mySymbol = Symbol();  
// 第一种写法  
let a = {};  
a[mySymbol] = 'Hello!';  
// 第二种写法  
let a = {  
  [mySymbol]: 'Hello!'  
};  
// 第三种写法  
let a = {};  
Object.defineProperty(a, mySymbol, { value: 'Hello!' });  
// 以上写法都得到同样结果  
a[mySymbol] // "Hello!"
```

作为属性名的Symbol

- 注意, Symbol 值作为对象属性名时, 不能用点运算符。

```
const mySymbol = Symbol();
```

```
const a = {};
```

```
a.mySymbol = 'Hello!';
```

```
a[mySymbol] // undefined
```

```
a['mySymbol'] // "Hello!"
```

Symbol特点
