

对象属性类型的问题

对象的 key 只能是字符串类型或 Symbol 类型，否则将会调用 toString() 方法转换成字符串类型。

Symbol 方法里面的参数也会被转换成字符串，调用 toString() 方法转换。

例一：

```
let a = {}, b = '123', c = 123;
a[b] = 'b'; // a['123'] = 'b';
a[c] = 'c'; // a['123'] = 'c';
console.log(a[b]); // 'c'
```

上面代码最后打印 a[b] 相当于打印 a['123']，结果是 'c'。

例二：

```
let a = {}, b = Symbol('123'), c = Symbol('123');
a[b] = 'b'; // a[Symbol('123')] = 'b';
a[c] = 'c'; // a[Symbol('123')] = 'c';
console.log(a[b]); // 'b'
```

上面代码中 a 的 key 同是 Symbol('123')，但这两个完全不同，不相等。后面不会覆盖前面的。

```
Symbol('123') === Symbol('123') // false
```

例三：

```
let a = {}, b = {key: '123'}, c = {key: '456'};
a[b] = 'b'; // a['[object Object]'] = 'b';
a[c] = 'c'; // a['[object Object]'] = 'c';
console.log(a[b]); // 'c'
```

上面代码打印 a[b]，等同于打印 a['[object Object]']，结果是 'c'。

扩展：了解 Symbol()

```
let x1 = Symbol(1)
x1 // Symbol(1)

let x2 = Symbol('1')
x2 // Symbol(1)
```

```
let x3 = Symbol(true)
x3 // Symbol(true)

let x4 = Symbol(false)
x4 // Symbol(false)

let x5 = Symbol([1, 2])
x5 // Symbol(1,2)

let x6 = Symbol({a: 1, b: 2})
x6 // Symbol([object Object])

(true).toString() // 'true'

[1, 2].toString() // '1,2'

Symbol(1) === Symbol(1) // false
```