

função que você chama com o novo.) O valor retornado pelo símbolo () não é igual a nenhum outro símbolo ou outro valor. Você pode passar uma corda para symbol (), e essa string é usada quando seu símbolo é convertido para uma string. Mas este é apenas um auxílio de depuração: dois símbolos criados com O mesmo argumento de string ainda é diferente um do outro.

O ponto dos símbolos não é segurança, mas definir uma extensão segura Mecanismo para objetos JavaScript. Se você receber um objeto de terceiros código que você não controla e precisa adicionar alguns dos seus próprios propriedades para esse objeto, mas querem ter certeza de que suas propriedades não conflito com nenhuma propriedade que já possa existir no objeto, Você pode usar com segurança símbolos como nomes de propriedades. Se você fizer isso, você também pode estar confiante de que o código de terceiros não será acidentalmente Altere suas propriedades simbolicamente nomeadas. (Esse código de terceiros poderia, Obviamente, use o object.getownPropertySymbols () para descobrir Os símbolos que você está usando e podem alterar ou excluir seu propriedades. É por isso que os símbolos não são um mecanismo de segurança.)

#### 6.10.4 Operador de espalhamento

No ES2018 e mais tarde, você pode copiar as propriedades de um objeto existente em um novo objeto usando o "operador de espalhamento" ... dentro de um objeto literal:

```
deixe a posição = {x: 0, y: 0};
```

```
Let Dimensions = {Width: 100, Hight: 75};
```

```
Deixe rect = {... posicionar, ... dimensões};
```

```
rect.x + rect.y + rect.width + rect.height // => 175
```

Neste código, as propriedades da posição e dimensões