um símbolo
Seja o = {};// Crie um novo objeto
o [strname] = 1;// Defina uma propriedade com um
Nome da string
o [symname] = 2;// Defina uma propriedade com um
Nome do símbolo
o [strname] // => 1: Acesse a string-
propriedade nomeada
o [symname] // => 2: Acesse o símbolo-
propriedade nomeada
O tipo de símbolo não possui uma sintaxe literal.Para obter um símbolo
valor, você chama a função Symbol ().Esta função nunca retorna
O mesmo valor duas vezes, mesmo quando chamado com o mesmo argumento.Esse
significa que, se você chama Symbol () para obter um valor de símbolo, você pode
use com segurança esse valor como nome de propriedade para adicionar uma nova propriedade a um
objeto e não precisa se preocupar com o fato de você estar substituindo um
Propriedade existente com o mesmo nome.Da mesma forma, se você usa simbólico
nomes de propriedades e não compartilham esses símbolos, você pode estar confiante
que outros módulos de código em seu programa não acidentalmente
substitua suas propriedades.
Na prática, os símbolos servem como um mecanismo de extensão de linguagem.Quando
O ES6 introduziu os objetos for/of loop (§5.4.4) e iterável
(Capítulo 12), precisava definir o método padrão que as classes poderiam
implementar para se tornar iterável.Mas padronizando qualquer
nome de string específico para este método de iterador teria quebrado
Código existente, então um nome simbólico foi usado.Como veremos em
Capítulo 12, Symbol.iterator é um valor de símbolo que pode ser usado
como um nome de método para tornar um objeto iterável.
A função Symbol () pega um argumento de string opcional e retorna