

um valor único de símbolo. Se você fornecer um argumento de string, essa string irá ser incluído na saída do método ToString () do símbolo.

Observe, no entanto, que chamando símbolo () duas vezes com a mesma corda produz dois valores de símbolo completamente diferentes.

Seja s = símbolo ("sym_x");

s.toString () // => "Symbol (sym_x)"

ToString () é o único método interessante de instâncias de símbolos.

Existem duas outras funções relacionadas ao símbolo que você deve conhecer,

no entanto. Às vezes, ao usar símbolos, você deseja mantê-los

privado para seu próprio código para que você tenha uma garantia de que suas propriedades

nunca entrará em conflito com as propriedades usadas por outro código. Outras vezes,

No entanto, você pode querer definir um valor de símbolo e compartilhá-lo amplamente

com outro código. Este seria o caso, por exemplo, se você fosse

Definindo algum tipo de extensão que você queria que outro código fosse capaz

Para participar, como no mecanismo de símbolo.iterator

descrito anteriormente.

Para servir este último caso de uso, JavaScript define um símbolo global

registro. A função symbol.for () leva um argumento de string e

Retorna um valor de símbolo associado à string que você passa. Se não

O símbolo já está associado a essa string, então um novo é criado

e retornou; Caso contrário, o símbolo já existente é retornado. Que

é, a função símbolo.for () é completamente diferente do

Symbol () Função: Symbol () nunca retorna o mesmo valor duas vezes,

mas symbol. para () sempre retorna o mesmo valor quando chamado com

a mesma corda. A string passada para symbol.for () aparece no

saída de toString () para o símbolo retornado, e também pode ser