

As propriedades são definidas como constantes nessa biblioteca. Se você está escrevendo código para criar os objetos que serão passados ?? para essa biblioteca, você pode Hardcode os nomes de propriedades, mas você corre o risco de bugs se digitar o Nome da propriedade Errado em qualquer lugar e você arriscará a incompatibilidade da versão Problemas se uma nova versão da biblioteca alterar a propriedade necessária nomes. Em vez disso, você pode achar que torna seu código mais robusto para Use a sintaxe de propriedade computada com as constantes de nome da propriedade definido pela biblioteca.

6.10.3 Símbolos como nomes de propriedades

A sintaxe da propriedade computada permite um outro objeto muito importante característica literal. No ES6 e posterior, os nomes de propriedades podem ser strings ou símbolos. Se você atribuir um símbolo a uma variável ou constante, então você pode Use esse símbolo como um nome de propriedade usando a propriedade computada sintaxe:

```
Extensão const = símbolo ("meu símbolo de extensão");  
Seja o = {  
[Extensão]: { / * Dados de extensão armazenados neste objeto * /  
}  
};  
o [extensão] .x = 0; // Isso não vai conflitar com outro
```

Propriedades de O.

Conforme explicado no §3.6, os símbolos são valores opacos. Você não pode fazer qualquer coisa com eles além de usá -los como nomes de propriedades. Todo Símbolo é diferente de todos os outros símbolos, no entanto, o que significa que os símbolos são bons para criar nomes exclusivos de propriedades. Crie a novo símbolo chamando a função de fábrica de símbolo (). (Símbolos são valores primitivos, não objetos, então símbolo () não é um construtor