função especificada.A sintaxe abreviada deixa mais claro essa área ()
é um método e não uma propriedade de dados como o lado.
Quando você escreve um método usando esta sintaxe abreviada, a propriedade
o nome pode assumir qualquer um dos formulários que são legais em um objeto literal:
Além de um identificador javascript regular, como a área de nome acima,
Você também pode usar literais de cordas e nomes de propriedades computadas, que
pode incluir nomes de propriedades de símbolo:

```
const métod_name = "m";
const símbolo = símbolo ();
Deixe WeirdMethods = {
"Método com espaços" (x) {return x + 1;},
[Method_name] (x) {return x + 2;},
[símbolo] (x) {return x + 3;}
};
WeirdMethods ["Método com espaços"] (1) // => 2
WeirdMethods [Method_Name] (1) // => 3
WeirdMethods [símbolo] (1) // => 4
```

Usar um símbolo como nome de método não é tão estranho quanto parece.Em
a fim de tornar um objeto iterável (para que possa ser usado com um para/de
loop), você deve definir um método com o nome simbólico
Symbol.iterator, e há exemplos de fazer exatamente isso em
Capítulo 12.

6.10.6 Getters de propriedades e setters

Todas as propriedades do objeto que discutimos até agora neste capítulo têm
foram propriedades de dados com um nome e um valor comum.JavaScript
também suporta propriedades de acessador, que não têm um único valor, mas
Em vez disso, tenha um ou dois métodos de acessórios: um getter e/ou um setter.