

Existem duas escolas de pensamento sobre o uso da palavra -chave const. Um  
A abordagem é usar const apenas para valores que são fundamentalmente imutáveis, como  
as constantes físicas mostradas, ou números de versão do programa, ou sequências de bytes  
usado para identificar tipos de arquivo, por exemplo. Outra abordagem reconhece que muitos de  
As chamadas variáveis ??em nosso programa nunca mudam como nosso programa  
corre. Nesta abordagem, declaramos tudo com const, e então se encontrarmos isso  
Na verdade, queremos permitir que o valor varie, mudamos a declaração para deixar.  
Isso pode ajudar a prevenir bugs descartando mudanças acidentais para variáveis ??que nós  
não pretendia.

Em uma abordagem, usamos const apenas para valores que não devem mudar. No outro,  
Usamos o const para qualquer valor que não mude. Eu prefiro o primeiro  
abordagem em meu próprio código.

No capítulo 5, aprenderemos sobre o para, para/in e para/de loop  
Declarações em JavaScript. Cada um desses loops inclui uma variável de loop  
Isso recebe um novo valor atribuído a ele em cada iteração do loop.

O JavaScript nos permite declarar a variável de loop como parte do loop  
sintaxe em si, e essa é outra maneira comum de usar LET:

```
para (vamos i = 0, len = data.length; i < len; i ++)
```

```
console.log (dados [i]);
```

```
para (Let Datum of Data) console.log (datum);
```

```
para (deixe a propriedade no objeto) console.log (propriedade);
```

Pode parecer surpreendente, mas você também pode usar o const para declarar o loop

?Variáveis? para/in e para/de loops, desde que o corpo do

O loop não reatribui um novo valor. Nesse caso, a declaração const

está apenas dizendo que o valor é constante durante a duração de um loop  
iteração:

```
para (const Datum of Data) console.log (datum);
```