

Próximo:[Observando variáveis](#), Anterior:[Acessando variáveis](#), Acima:[Variáveis](#) [Conteúdo][Índice]

12.8 Configurando Valores Variáveis

A maneira usual de alterar o valor de uma variável é com o formulário especial `setq`. Quando você precisar calcular a escolha da variável em tempo de execução, use a função `set`.

Forma Especial: `setq` [*forma do símbolo*]...

Essa forma especial é o método mais comum de alterar o valor de uma variável. Cada *símbolo* recebe um novo valor, que é o resultado da avaliação da *forma* correspondente. A ligação atual do símbolo é alterada.

`setq` não avalia *símbolo*; ele define o símbolo que você escreve. Dizemos que este argumento é *automaticamente citado*. O 'q' em `setq` significa "citado".

O valor do `setq` formulário é o valor do último *formulário*.

```
(conjunto x (1+ 2))
⇒ 3
x; x agora tem um valor global.
⇒ 3
(deixe ((x 5))
  (setq x 6); A ligação local de x está definida.
  x)
⇒ 6
x; O valor global permanece inalterado.
⇒ 3
```

Observe que o primeiro *formulário* é avaliado, o primeiro *símbolo* é definido, o segundo *formulário* é avaliado, o segundo *símbolo* é definido e assim por diante:

```
(setq x 10 ; Observe que x é definido antes de
      y (1+ x)); o valor de y é calculado.
⇒ 11
```

Função: definir o valor do símbolo

Esta função coloca *valor* na célula de valor do *símbolo*. Como é uma função e não uma forma especial, a expressão escrita para *símbolo* é avaliada para obter o símbolo a ser definido. O valor de retorno é *valor*.

Quando a vinculação de variável dinâmica está em vigor (o padrão), `set` tem o mesmo efeito que `setq`, exceto pelo fato de `set` avaliar seu argumento de *símbolo*/`setq`, mas não. Mas quando uma variável é vinculada lexicalmente, `set` afeta seu valor *dinâmico*, enquanto `setq` afeta seu valor atual (lexical). Consulte [Escopo variável](#).

```
(defina um 1)
error→ O valor do símbolo como variável é nulo: um
(defina 'um 1)
⇒ 1
```

```
(definir 'dois' um)
  => um
(conjunto dois 2); two avalia como símbolo one.
  => 2
1 ; Então é isso one que foi definido.
  => 2
(let ((one 1)) ; Esta ligação de one é definida,
  (set 'one 3) ;    não o valor global.
1)
  => 3
1
  => 2
```

Se o símbolo não for realmente um símbolo, um wrong-type-argument erro é sinalizado.

```
(conjunto '(xy) 'z)
erro→ tipo de argumento errado: symbolp, (xy)
```

Próximo:[Observando variáveis](#), Anterior:[Acessando variáveis](#), Acima:[Variáveis](#) [Conteúdo][Índice]