

Próximo: [Definindo Variáveis](#), Anterior: [Variáveis locais](#), Acima: [Variáveis](#) [[Conteúdo](#)][[Índice](#)]

## 12.4 Quando uma variável é nula

Dizemos que uma variável é nula se seu símbolo tiver uma célula de valor não atribuída (consulte [Componentes de símbolo](#) ).

Sob a regra de escopo dinâmico padrão do Emacs Lisp (consulte [Variable Scoping](#) ), a célula de valor armazena o valor atual (local ou global) da variável. Observe que uma célula de valor não atribuída *não* é o mesmo que ter `nil` na célula de valor. O símbolo `nil` é um objeto Lisp e pode ser o valor de uma variável, assim como qualquer outro objeto pode ser; mas ainda é um valor. Se uma variável for nula, tentar avaliar a variável sinaliza um `void-variable` erro, em vez de retornar um valor.

Sob a regra de escopo léxico opcional, a célula de valor contém apenas o valor global da variável — o valor fora de qualquer construção de ligação léxica. Quando uma variável é vinculada lexicalmente, o valor local é determinado pelo ambiente léxico; portanto, as variáveis podem ter valores locais mesmo se as células de valor de seus símbolos não estiverem atribuídas.

### Função: *símbolo* `makunbound`

Esta função esvazia a célula de valor de *symbol* , tornando a variável nula. Ele retorna o *símbolo* .

Se o *símbolo* tiver uma vinculação local dinâmica, `makunbound` anulará a vinculação atual e essa vacuidade durará apenas enquanto a vinculação local estiver em vigor. Depois, a ligação local ou global anteriormente sombreada é reexposta; então a variável não será mais nula, a menos que a ligação reexposta também seja nula.

Aqui estão alguns exemplos (assumindo que a vinculação dinâmica está em vigor):

```
(setq x 1); Coloque um valor na ligação global.
⇒ 1
(let ((x 2)) ; Ligar localmente.
  (makunbound 'x) ; Anular a ligação local.
  x)
error→ 0 valor do símbolo como variável é nulo: x
x; A ligação global permanece inalterada.
⇒ 1

(let ((x 2)) ; Ligar localmente.
  (let ((x 3)) ; E novamente.
    (makunbound 'x) ; Anular a ligação local mais interna.
    x)) ; E consulte: é nulo.
error→ 0 valor do símbolo como variável é nulo: x

(deixe ((x 2))
  (deixe ((x 3))
    (makunbound 'x)); Anule a encadernação interna e, em seguida, remova-a
    x); Agora a ligação externa leté visível.
⇒ 2
```

### Função: *variável* `boundp`

Esta função retorna `t` se a *variável* (um símbolo) não for nula e `nil` se for nula.

Aqui estão alguns exemplos (assumindo que a vinculação dinâmica está em vigor):

```
(boundp 'abracadabra); Começa vazio.  
⇒ nada  
(let ((abracadabra 5)) ; Liga-o localmente.  
  (boundp 'abracadabra))  
⇒ t  
(boundp 'abracadabra); Ainda globalmente nulo.  
⇒ nada  
(setq abracadabra 5); Torná-lo globalmente não nulo.  
⇒ 5  
(boundp 'abracadabra)  
⇒ t
```

Próximo: [Definindo Variáveis](#), Anterior: [Variáveis locais](#), Acima: [Variáveis](#) [[Conteúdo](#)][[Índice](#)]