

Anterior:[Ligação Lexical](#), Acima:[Escopo Variável](#) [[Conteúdo](#)][[Índice](#)]

12.10.4 Usando Ligação Lexical

Ao carregar um arquivo Lisp do Emacs ou avaliar um buffer Lisp, a vinculação léxica é habilitada se a variável `buffer-local lexical-binding` for não-`nil`:

Variável: `lexical-binding`

Se essa variável local de buffer for não-`nil`, os arquivos e buffers do Emacs Lisp serão avaliados usando a associação léxica em vez da associação dinâmica. (Entretanto, variáveis especiais ainda são vinculadas dinamicamente; veja abaixo.) Se `nil`, a vinculação dinâmica é usada para todas as variáveis locais. Essa variável é normalmente definida para um arquivo Emacs Lisp inteiro, como uma variável local de arquivo (consulte [Variáveis locais de arquivo](#)). Observe que, ao contrário de outras variáveis, esta deve ser definida na primeira linha de um arquivo.

Ao avaliar o código Emacs Lisp diretamente usando uma `eval` chamada, a associação léxica é habilitada se o argumento *léxicoeval* para não for `nil`. Veja [Eval](#).

A vinculação léxica também é habilitada no modo Lisp Interaction e IELM, usado no*coçar, arranhão*e*ielm*buffers, e também ao avaliar expressões via `M-:(eval-expression)` e ao processar o-`-eval` opções de linha de comando do Emacs (consulte [Argumentos de ação](#) no Manual do GNU Emacs) e `emacsclient` (consulte [Opções do emacsclient](#) no Manual do GNU Emacs).

Mesmo quando a vinculação léxica estiver habilitada, certas variáveis continuarão sendo vinculadas dinamicamente. Estas são chamadas *de variáveis especiais*. Cada variável que foi definida com `defvar`, `defcustom` ou `defconst` é uma variável especial (consulte [Definindo Variáveis](#)). Todas as outras variáveis estão sujeitas a vinculação léxica.

Usando `defvar` sem um valor, é possível vincular uma variável dinamicamente apenas em um arquivo ou em apenas uma parte de um arquivo enquanto ainda a vincula lexicalmente em outro lugar. Por exemplo:

```
(deixar (_)
  (defvar x); As ligações de xlet serão dinâmicas dentro deste let.
  (let ((x -99)) ; Esta é uma ligação dinâmica de x.
    (defun get-dynamic-x()
      x)))

(let ((x 'lexical)) ; Esta é uma ligação léxica de x.
  (defun get-lexical-x()
    x))

(deixar (_)
  (defvar x)
  (deixe ((x 'dinâmico))
    (lista (get-lexical-x)
            (get-dinâmico-x))))
⇒ (dinâmica lexical)
```

Função: *símbolo de variável especial-p*

Esta função retorna o *símbolo* `nil` não- se é uma variável especial (ou seja, tem uma definição de variável, , ou). Caso contrário, o valor de retorno é `. defvardefcustomdefconstnil`

Observe que, como esta é uma função, ela só pode retornar `non-nil` para variáveis que são permanentemente especiais, mas não para aquelas que são apenas especiais no escopo lexical atual.

O uso de uma variável especial como argumento formal em uma função é desencorajado. Isso dá origem a um comportamento não especificado quando o modo de associação léxica está ativado (pode usar associação léxica às vezes e associação dinâmica outras vezes).

Converter um programa Emacs Lisp para vinculação léxica é fácil. Primeiro, adicione uma configuração de variável local de arquivo de `lexical-binding` na linha de cabeçalho do arquivo de origem do Emacs Lisp (consulte [Variáveis locais de arquivo](#)). Segundo, verifique se cada variável no programa que precisa ser vinculada dinamicamente tem uma definição de variável, para que não seja inadvertidamente vinculada lexicalmente.

Uma maneira simples de descobrir quais variáveis precisam de uma definição de variável é compilar por byte o arquivo de origem. Consulte [Compilação de bytes](#). Se uma variável não especial for usada fora de um `let` formulário, o compilador de bytes avisará sobre referência ou atribuição a uma variável livre. Se uma variável não especial estiver vinculada, mas não for usada em um `let` formulário, o compilador de bytes avisará sobre uma variável léxica não utilizada. O compilador de bytes também emitirá um aviso se você usar uma variável especial como argumento de função.

(Para silenciar os avisos do compilador de bytes sobre variáveis não utilizadas, basta usar um nome de variável que comece com um sublinhado. O compilador de bytes interpreta isso como uma indicação de que esta é uma variável conhecida por não ser usada.)

Anterior: [Ligação Lexical](#), Acima: [Escopo Variável](#) [[Conteúdo](#)][[Índice](#)]