

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

12.10.2 Uso Adequado da Vinculação Dinâmica

A vinculação dinâmica é um recurso poderoso, pois permite que os programas se refiram a variáveis que não estão definidas em seu escopo textual local. No entanto, se usado sem restrições, isso também pode tornar os programas difíceis de entender. Existem duas maneiras limpas de usar essa técnica:

- Se uma variável não tiver uma definição global, use-a como uma variável local apenas em uma construção de associação, como o corpo do `let` formulário em que a variável foi vinculada. Se esta convenção for seguida consistentemente em todo o programa, o valor da variável não afetará nem será afetado por nenhum uso do mesmo símbolo de variável em qualquer outro lugar do programa.
- Caso contrário, defina a variável com `defvar`, `defconst` (consulte [Definindo variáveis](#)) ou `defcustom` (consulte [Definições de variáveis](#)). Normalmente, a definição deve estar no nível superior em um arquivo Emacs Lisp. Na medida do possível, deve incluir uma string de documentação que explique o significado e o propósito da variável. Você também deve escolher o nome da variável para evitar conflitos de nome (consulte [Convenções de codificação](#)).

Então você pode vincular a variável em qualquer lugar em um programa, sabendo com segurança qual será o efeito. Onde quer que você encontre a variável, será fácil consultar a definição, por exemplo, através do `C-h v` comando (desde que a definição da variável tenha sido carregada no Emacs). Veja a [Ajuda de Nome](#) no Manual do GNU Emacs .

Por exemplo, é comum usar ligações locais para variáveis personalizáveis como `case-fold-search`:

```
(defun search-for-abc ()  
  "Procure pela string \"abc\", ignorando as diferenças entre maiúsculas  
  (seja ((caso-dobra-pesquisa t)))  
  (pesquisar novamente \"abc\"))
```

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