

Próximo:[Combinadores de conselhos](#), Anterior:[Primitivos principais de aconselhamento](#), Acima:

[Funções de aconselhamento](#) [[Conteúdo](#)][[Índice](#)]

13.11.2 Aconselhando Funções Nomeadas

Um uso comum de aconselhamento é para funções nomeadas e macros. Você poderia simplesmente usar `add-function` como em:

```
(add-function :around (symbol-function ' fun ) #'his-tracing-function)
```

Mas você deve usar `advice-add` e `advice-remove` para isso em vez disso. Este conjunto separado de funções para manipular conselhos aplicados a funções nomeadas oferece os seguintes recursos extras em comparação com `add-function`: eles sabem como lidar com macros e funções carregadas automaticamente, eles permitem `describe-function` preservar a docstring original, bem como documentar os conselhos adicionados, e eles permitem adicionar e remover conselhos antes mesmo de uma função ser definida.

`advice-add` pode ser útil para alterar o comportamento de chamadas existentes para uma função existente sem ter que redefinir toda a função. No entanto, pode ser uma fonte de bugs, pois os chamadores existentes para a função podem assumir o comportamento antigo e funcionar incorretamente quando o comportamento é alterado por aviso. O conselho também pode causar confusão na depuração, se a pessoa que está fazendo a depuração não perceber ou lembrar que a função foi modificada pelo conselho.

Por esses motivos, o conselho deve ser reservado para os casos em que você não pode modificar o comportamento de uma função de nenhuma outra forma. Se for possível fazer a mesma coisa por meio de um gancho, é preferível (consulte [Hooks](#)). Se você simplesmente deseja alterar o que uma determinada tecla faz, pode ser melhor escrever um novo comando e remapear as combinações de teclas do comando antigo para a nova (consulte [Remapeamento de comandos](#)).

Se você estiver escrevendo código para lançamento, para outros usarem, tente evitar incluir conselhos nele. Se a função que você deseja aconselhar não tiver gancho para fazer o trabalho, converse com os desenvolvedores do Emacs sobre como adicionar um gancho adequado. Especialmente, os próprios arquivos de origem do Emacs não devem colocar conselhos sobre funções no Emacs. (Existem atualmente algumas exceções a essa convenção, mas pretendemos corrigi-las.) Geralmente é mais limpo criar um novo gancho em `foo`, e fazer baruso do gancho, do que barcolocar conselhos em `foo`.

Formulários especiais (veja [Formulários Especiais](#)) não podem ser aconselhados, entretanto macros podem ser aconselhadas, da mesma forma que funções. Obviamente, isso não afetará o código que já foi expandido por macro, portanto, você precisa certificar-se de que o aviso esteja instalado antes que a macro seja expandida.

É possível aconselhar uma primitiva (consulte [O que é uma função](#)), mas normalmente *não* se deve fazê-lo, por dois motivos. Em primeiro lugar, algumas primitivas são usadas pelo mecanismo de aconselhamento, e aconselhá-las pode causar uma recursão infinita. Em segundo lugar, muitas primitivas são chamadas diretamente de C, e tais chamadas ignoram os avisos; portanto, acaba-se em uma situação confusa onde algumas chamadas (ocorrendo do código Lisp) obedecem ao conselho e outras chamadas (do código C) não.

Macro: símbolo de definição de conselho (*onde lista lambda e profundidade de nome opcional*) & resto do corpo

Essa macro define um conselho e o adiciona à função chamada *symbol* . O conselho é uma função anônima se *name* for nil ou uma função chamada *symbol@name*. Veja *advice-add* para explicação de outros argumentos.

Função: símbolo de adição de conselho onde função e adereços opcionais

Adicione a função de aconselhamento ao símbolo de função nomeado . *onde* e *props* têm o mesmo significado que para *add-function* (veja [Primitivos](#) de Orientação Básica).

Função: função de símbolo de remoção de conselho

Remova a função de aconselhamento do símbolo de função nomeado . A função também pode ser a nome de um conselho.

Função: símbolo de função conselheiro-membro-p

Retorne não- se a *função* nil de aconselhamento já estiver no símbolo da função nomeada . A função também pode ser a de um conselho. *name*

Função: símbolo de função de conselho-mapc

Chame a função para cada conselho que foi adicionado ao símbolo da função nomeada . A função é chamada com dois argumentos: a função de aconselhamento e suas propriedades.

Próximo: [Combinadores de conselhos](#), Anterior: [Primitivos principais de aconselhamento](#), Acima:

[Funções de aconselhamento](#) [[Conteúdo](#)][[Índice](#)]