

Anterior:[Lista de argumentos](#), Acima:[Expressões lambda](#) [[Conteúdo](#)][[Índice](#)]

13.2.4 Documentação de Cadeias de Funções

Uma expressão lambda pode opcionalmente ter uma *string de documentação* logo após a lista lambda. Essa string não afeta a execução da função; é um tipo de comentário, mas um comentário sistematizado que realmente aparece dentro do mundo Lisp e pode ser usado pelas facilidades de ajuda do Emacs. Consulte [Documentação](#) , para saber como a string de documentação é acessada.

É uma boa idéia fornecer strings de documentação para todas as funções em seu programa, mesmo aquelas que são chamadas apenas de dentro de seu programa. As strings de documentação são como comentários, exceto que são mais fáceis de acessar.

A primeira linha da string de documentação deve ser independente, porque a `proposexibe` apenas essa primeira linha. Deve consistir em uma ou duas frases completas que resumam o propósito da função.

O início da string de documentação geralmente é recuado no arquivo de origem, mas como esses espaços vêm antes das aspas duplas iniciais, eles não fazem parte da string. Algumas pessoas praticam recuar quaisquer linhas adicionais da string para que o texto se alinhe na fonte do programa. *Isso é um erro*. O recuo das linhas a seguir está dentro da string; o que parece bom no código-fonte ficará feio quando exibido pelos comandos de ajuda.

Você pode se perguntar como a string de documentação pode ser opcional, já que existem componentes obrigatórios da função que a seguem (o corpo). Como a avaliação de uma string retorna essa string, sem nenhum efeito colateral, ela não tem efeito se não for a última forma no corpo. Assim, na prática, não há confusão entre a primeira forma do corpo e a string de documentação; se a única forma do corpo for uma string, ela servirá tanto como valor de retorno quanto como documentação.

A última linha da string de documentação pode especificar convenções de chamada diferentes dos argumentos reais da função. Escreva um texto assim:

```
\(fn arglist )
```

seguindo uma linha em branco, no início da linha, sem nova linha dentro da string de documentação. (O `'\'` é usado para evitar confundir os comandos de movimento do Emacs.) A convenção de chamada especificada dessa maneira aparece nas mensagens de ajuda no lugar daquela derivada dos argumentos reais da função.

Esse recurso é particularmente útil para definições de macro, pois os argumentos escritos em uma definição de macro geralmente não correspondem à maneira como os usuários pensam nas partes da chamada de macro.

Não use esse recurso se desejar descontinuar a convenção de chamada e favorecer a que você anuncia pela especificação acima. Em vez disso, use a `advertised-calling-convention` declaração (consulte [Declare Form](#)) ou `set-advertised-calling-convention` (consulte [Obsolete Functions](#)), porque esses dois farão com que o compilador de bytes emita uma mensagem de aviso quando compilar programas Lisp que usam a convenção de chamada obsoleta.

Anterior:[Lista de argumentos](#), Acima:[Expressões lambda](#) [[Conteúdo](#)][[Índice](#)]

