

Próximo:[Declarando Funções](#), Anterior:[Funções embutidas](#), Acima:[Funções](#) [[Conteúdo](#)][[Índice](#)]

13.14 O `declareFormulário`

`declare` é uma macro especial que pode ser usada para adicionar propriedades meta a uma função ou macro: por exemplo, marcando-a como obsoleta, ou dando a seus formulários uma convenção especial de recuo TAB no modo Emacs Lisp.

Macro: declarar *especificações*...

Essa macro ignora seus argumentos e avalia como `nil`; não tem efeito de tempo de execução. No entanto, quando um `declareFormulário` ocorre no argumento *declaredefun* de uma definição de função ou `defsubst` (consulte [Definindo funções](#)) ou uma `defmacro` definição de macro (consulte [Definindo macros](#)), ele anexa as propriedades especificadas por *especificações* à função ou macro. Este trabalho é especialmente realizado por `defun`, `defsubst`, e `defmacro`.

Cada elemento nas *especificações* deve ter o formulário , que não deve ser citado. Estes têm os seguintes efeitos: (*property args...*)

(**advertised-calling-convention** *signature when*)

Isso funciona como uma chamada para `set-advertised-calling-convention` (veja [Obsolete Functions](#)); *assinatura* especifica a lista de argumentos correta para chamar a função ou macro e *quando* deve ser uma string indicando quando a lista de argumentos antiga se tornou obsoleta.

(**debug** *edebug-form-spec*)

Isso é válido apenas para macros. Ao percorrer a macro com Edebug, use *edebug-form-spec*. Consulte [Instrumentação de chamadas de macro](#).

(**doc-string** *n*)

Isso é usado ao definir uma função ou macro que será usada para definir entidades como funções, macros ou variáveis. Indica que o *enésimo* argumento, se houver, deve ser considerado como uma string de documentação.

(**indent** *indent-spec*)

Recuar chamadas para esta função ou macro de acordo com *indent-spec*. Isso geralmente é usado para macros, embora funcione também para funções. Consulte [Recuo de macros](#).

(**interactive-only** *value*)

Defina a `interactive-only` propriedade da função para *value*. Consulte [A propriedade somente interativa](#).

(**obsolete** *current-name when*)

Marque a função ou macro como obsoleta, semelhante a uma chamada para `make-obsolete` (consulte [Funções obsoletas](#)). *nome-corrente* deve ser um símbolo (nesse caso, a mensagem de aviso diz para usar isso), uma string (especificando a mensagem de aviso) ou `nil` (nesse caso, a mensagem de aviso não fornece detalhes extras). *when* deve ser uma string indicando quando a função ou macro se tornou obsoleta pela primeira vez.

(**compiler-macro** *expander*)

Isso só pode ser usado para funções e informa ao compilador para usar o *expansor* como uma função de otimização. Ao encontrar uma chamada para a função, do formulário , o expansor de macro chamará *expansor* com esse formulário, bem como com *args ...*, e o *expansor* pode retornar uma nova expressão para usar em vez da chamada de função ou pode retornar apenas o forma inalterada, para indicar que a chamada da função deve ser deixada sozinha. *expander* pode ser um símbolo, ou pode ser uma forma em que *arg* conterá a expressão de chamada de função original e os argumentos (não avaliados) para a função podem ser acessados usando os argumentos formais da função. (*function args...*) (lambda (*arg*) *body*)

(gv-expander *expander*)

Declare *expansor* para ser a função para manipular chamadas para a macro (ou função) como uma variável generalizada, de forma semelhante a *gv-define-expander*. *expander* pode ser um símbolo ou pode ser da forma em que, nesse caso, essa função terá acesso adicional aos argumentos da macro (ou função). (lambda (*arg*) *body*)

(gv-setter *setter*)

Declare *setter* para ser a função para manipular chamadas para a macro (ou função) como uma variável generalizada. *setter* pode ser um símbolo, caso em que será passado para *gv-define-simple-setter*, ou pode ser da forma em que a função terá acesso adicionalmente aos argumentos da macro (ou função) e será passada para . (lambda (*arg*) *body*)*gv-define-setter*

Próximo:[Declarando Funções](#), Anterior:[Funções embutidas](#), Acima:[Funções](#) [[Conteúdo](#)][[Índice](#)]