

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

13.12 Declarando Funções Obsoletas

Você pode marcar uma função nomeada como *obsoleta*, o que significa que ela pode ser removida em algum momento no futuro. Isso faz com que o Emacs avise que a função está obsoleta sempre que compilar byte o código que contém essa função e sempre que exibir a documentação dessa função. Em todos os outros aspectos, uma função obsoleta se comporta como qualquer outra função.

A maneira mais fácil de marcar uma função como obsoleta é colocar um `(declare (obsolete ...))` formulário na definição da função. Consulte [Formulário de declaração](#). Alternativamente, você pode usar a `make-obsolete` função, descrita abaixo.

Uma macro (veja [Macros](#)) também pode ser marcada como obsoleta com `make-obsolete`; isso tem os mesmos efeitos que para uma função. Um alias para uma função ou macro também pode ser marcado como obsoleto; isso torna o próprio alias obsoleto, não a função ou macro para a qual ele resolve.

Função: tornar obsoleto *nome-obsoleto nome-atual quando*

Esta função marca o *nome* obsoleto como obsoleto. *obsolete-name* deve ser um símbolo que nomeia uma função ou macro, ou um alias para uma função ou macro.

Se *current-name* for um símbolo, a mensagem de aviso diz para usar *current-name* em vez de *obsolete-name*. *nome-corrente* não precisa ser um alias para *nome-obsoleto*; pode ser uma função diferente com funcionalidade semelhante. *current-name* também pode ser uma string, que serve como mensagem de aviso. A mensagem deve começar em letras minúsculas e terminar com um ponto. Também pode ser `nil`, caso em que a mensagem de aviso não fornece detalhes adicionais.

O argumento *when* deve ser uma string indicando quando a função se tornou obsoleta pela primeira vez—por exemplo, uma data ou um número de lançamento.

Macro: define-obsolete-function-alias *nome-obsoleto nome-atual quando &documento opcional*

Essa macro de conveniência marca a função *obsolete-name* obsoleta e também a define como um alias para a função *current-name*. É equivalente ao seguinte:

```
(defalias obsolete-name nome -corrente doc)
(tornar obsoleto nome obsoleto nome atual quando)
```

Além disso, você pode marcar uma convenção de chamada específica para uma função como obsoleta:

Função: assinatura da função set-advertised-calling-convention quando

Esta função especifica a assinatura da lista de argumentos como a maneira correta de chamar *function*. Isso faz com que o compilador de bytes do Emacs emita um aviso sempre que se deparar com um programa Emacs Lisp que chama a *função* de qualquer outra maneira (no entanto, ele ainda permitirá que o código seja compilado por byte). *when* deve ser uma string indicando quando a variável se tornou obsoleta pela primeira vez (geralmente uma string de número de versão).

Por exemplo, em versões antigas do Emacs a `sit-for` função aceitava três argumentos, como este

```
(sit-for segundos milissegundos nodisp)
```

No entanto, chamar `sit-for` dessa maneira é considerado obsoleto (consulte [Waiting](#)). A antiga convenção de chamada está obsoleta assim:

```
(definir-convenção-chamada anunciada  
  'sit-for '(segundos e nodisp opcional) "22.1")
```

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