

Próximo:[Definições de grupo](#), Acima:[Customização](#) [[Conteúdo](#)][[Índice](#)]

15.1 Palavras-chave de itens comuns

As declarações de personalização que descreveremos nas próximas seções— `defcustom`, `defgroup`, etc.— todas aceitam argumentos de palavras-chave (consulte [Variáveis constantes](#)) para especificar várias informações. Esta seção descreve palavras-chave que se aplicam a todos os tipos de declarações de personalização.

Todas essas palavras-chave, exceto `:tag`, podem ser usadas mais de uma vez em um determinado item. Cada uso da palavra-chave tem um efeito independente. A palavra-chave `:tag` é uma exceção porque qualquer item pode exibir apenas um nome.

`:tag` *label*

Use *label* , uma string, em vez do nome do item, para rotular o item em menus de personalização e buffers. **Não use uma etiqueta que seja substancialmente diferente do nome real do item; isso causaria confusão.**

`:group` *group*

Coloque este item de personalização no grupo *group* . Se esta palavra-chave estiver faltando em um item de personalização, ela será colocada no mesmo grupo que foi definido por último (no arquivo atual).

Quando você usa `:group` em a `defgroup`, torna o novo grupo um subgrupo de *grupo* .

Se você usar essa palavra-chave mais de uma vez, poderá colocar um único item em mais de um grupo. A exibição de qualquer um desses grupos mostrará este item. Por favor, não exagere, pois o resultado seria irritante.

`:link` *link-data*

Inclua um link externo após a string de documentação para este item. Esta é uma frase contendo um botão que faz referência a alguma outra documentação.

Existem várias alternativas que você pode usar para *link-data* :

`(custom-manual info-node)`

Link para um nó Info; *info-node* é uma string que especifica o nome do nó, como em " (emacs)Top". O link aparece como '[Manual]' no buffer de personalização e insere o leitor de informações integrado no *info-node* .

`(info-link info-node)`

Como `custom-manual`, exceto que o link aparece no buffer de customização com o nome do nó Info.

`(url-link url)`

Link para uma página da web; *url* é uma string que especifica a URL . O link aparece no buffer de personalização como *url* e chama o navegador WWW especificado por `browse-url-browser-function`.

`(emacs-commentary-link library)`

Link para a seção de comentários de uma biblioteca; *library* é uma string que especifica o nome da biblioteca. Consulte [Cabeçalhos da Biblioteca](#) .

(emacs-library-link *library*)

Link para um arquivo de biblioteca Emacs Lisp; *library* é uma string que especifica o nome da biblioteca.

(file-link *file*)

Link para um arquivo; *file* é uma string que especifica o nome do arquivo a ser visitado find-file quando o usuário invoca este link.

(function-link *function*)

Link para a documentação de uma função; *function* é uma string que especifica o nome da função a ser descrita describe-function quando o usuário invoca este link.

(variable-link *variable*)

Link para a documentação de uma variável; *variável* é uma string que especifica o nome da variável a ser descrita describe-variable quando o usuário invoca este link.

(custom-group-link *group*)

Link para outro grupo de personalização. Invocá-lo cria um novo buffer de personalização para *group* .

Você pode especificar o texto a ser usado no buffer de personalização adicionando após o primeiro elemento do *link-data* ; por exemplo, faz um link para o manual do Emacs que aparece no buffer como ':tag *name*(info-link :tag "foo" "(emacs)Top") foo'.

Você pode usar essa palavra-chave mais de uma vez para adicionar vários links.

:load *file*

Carregue o arquivo de *arquivo* (uma string) antes de exibir este item de personalização (consulte [Carregando](#)). O carregamento é feito com load, e somente se o arquivo ainda não estiver carregado.

:require *feature*

Execute quando suas personalizações salvas definirem o valor deste item. *recurso* deve ser um símbolo. (require '*feature*)

O motivo mais comum para usar :require é quando uma variável habilita um recurso como um modo secundário, e apenas definir a variável não terá nenhum efeito, a menos que o código que implementa o modo seja carregado.

:version *version*

Esta palavra-chave especifica que o item foi introduzido pela primeira vez na versão do Emacs *version* , ou que seu valor padrão foi alterado nessa versão. A *versão do* valor deve ser uma string.

:package-version '*(package . version)*

Essa palavra-chave especifica que o item foi introduzido pela primeira vez na versão *do pacote* *version* , ou que seu significado ou valor padrão foi alterado nessa versão. Esta palavra-chave tem prioridade sobre . :version

pacote deve ser o nome oficial do pacote, como um símbolo (por exemplo, MH-E). *version* deve ser uma string. Se o pacote *pacote* for lançado como parte do Emacs, *pacote* e *versão* devem aparecer no valor de `customize-package-emacs-version-alist`.

Os pacotes distribuídos como parte do Emacs que usam a palavra- `:package-version` chave também devem atualizar a `customize-package-emacs-version-alist` variável.

Variável: `customize-package-emacs-version-alist`

Essa lista fornece um mapeamento para as versões do Emacs associadas às versões de um pacote listado na palavra- `:package-version` chave. Seus elementos são:

```
( pacote ( pversion . everversion )...)
```

Para cada *pacote* , que é um símbolo, há um ou mais elementos que contêm uma versão do pacote *pversion* com uma versão do Emacs *everversion* associada . Essas versões são strings. Por exemplo, o pacote MH-E atualiza essa lista com o seguinte:

```
(adicionar à lista 'customize-package-emacs-version-alist
  '(MH-E ("6.0" . "22.1") ("6.1" . "22.1") ("7.0" . "22.1")
    ("7.1" . "22.1") ("7.2" . "22.1") ("7.3" . "22.1")
    ("7,4" . "22,1") ("8,0" . "22,1"))))
```

O valor do *pacote* precisa ser exclusivo e deve corresponder ao valor do *pacote* que aparece na palavra- `:package-version` chave. Como o usuário pode ver o valor em uma mensagem de erro, uma boa escolha é o nome oficial do pacote, como MH-E ou Gnus.

Próximo: [Definições de grupo](#), Acima: [Customização](#) [[Conteúdo](#)][[Índice](#)]