

Próximo:[Definindo novos tipos](#), Anterior:[Dividindo em listas](#), Acima:[Tipos de personalização](#) [Conteúdo][\[Índice\]](#)

15.4.4 Tipo de Palavras-chave

Você pode especificar pares de argumento-palavra-chave em um tipo de personalização após o símbolo do nome do tipo. Aqui estão as palavras-chave que você pode usar e seus significados:

:value default

Forneça um valor padrão.

Se `nil` não for um valor válido para a alternativa, é essencial especificar um padrão válido com `:value`.

Se você usar isso para um tipo que aparece como alternativa dentro de `choice`; especifica o valor padrão a ser utilizado, inicialmente, se e quando o usuário selecionar esta alternativa com o menu no buffer de customização.

Claro, se o valor real da opção se encaixar nesta alternativa, aparecerá mostrando o valor real, não o *padrão*.

:format format-string

Esta string será inserida no buffer para representar o valor correspondente ao tipo. A seguir '%' escapes estão disponíveis para uso em *format-string*:

'%[*botão* %]'

Exiba o *botão* de texto marcado como um botão. O `:action` atributo especifica o que o botão fará se o usuário o invocar; seu valor é uma função que recebe dois argumentos — o widget no qual o botão aparece e o evento.

Não há como especificar dois botões diferentes com ações diferentes.

'%{ *amostra* %}'

Mostrar *amostra* em uma face especial especificada por `:sample-face`.

'%v'

Substitua o valor do item. A forma como o valor é representado depende do tipo de item e (para variáveis) do tipo de personalização.

'%d'

Substitua a string de documentação do item.

'%h'

Como '%d', mas se a string de documentação for mais de uma linha, adicione um botão para controlar se deve mostrar tudo ou apenas a primeira linha.

'%t'

Substitua a tag aqui. Você especifica a tag com a palavra- `:tag` chave.

'%'

Exibir um literal '%'.

:action *action*

Execute a *ação* se o usuário clicar em um botão.

:button-face *face*

Use o rosto de *rosto* (um nome de rosto ou uma lista de nomes de rosto) para o texto do botão exibido com '%[...%]'.

:button-prefix *prefix*

:button-suffix *suffix*

Eles especificam o texto a ser exibido antes e depois de um botão. Cada um pode ser:

nil

Nenhum texto é inserido.

uma linha

A string é inserida literalmente.

um símbolo

O valor do símbolo é usado.

:tag *tag*

Use *tag* (uma string) como tag para o valor (ou parte do valor) que corresponde a esse tipo.

:doc *doc*

Use *doc* como a string de documentação para este valor (ou parte do valor) que corresponde a este tipo. Para que isso funcione, você deve especificar um valor para :formate usar '%d' ou '%h' nesse valor.

A razão usual para especificar uma string de documentação para um tipo é fornecer mais informações sobre os significados das alternativas dentro de um *choicetipo* ou as partes de algum outro tipo composto.

:help-echo *motion-doc*

Quando você move para este item com *widget-forward* ou *widget-backward*, ele exibirá o string *motion-doc* na área de eco. Além disso, *motion-doc* é usado como a string do mouse *help-echoe*. Pode ser uma função ou formulário avaliado para gerar uma string de ajuda. Se for uma função, ela é chamada com um argumento, o *widget*.

:match *function*

Especifique como decidir se um valor corresponde ao tipo. O valor correspondente, *function*, deve ser uma função que aceita dois argumentos, um *widget* e um valor; deve retornar non-*nil* se o valor for aceitável.

:match-inline *function*

Especifique como decidir se um valor embutido corresponde ao tipo. O valor correspondente, *function*, deve ser uma função que aceita dois argumentos, um *widget* e um valor embutido; deve

retornar non-`nil` se o valor for aceitável. Consulte [Emendando em listas](#) para obter mais informações sobre valores embutidos.

:validate *function*

Especifique uma função de validação para entrada. A *função* recebe um widget como argumento e deve retornar `nil` se o valor atual do widget for válido para o widget. Caso contrário, ele deve retornar o widget que contém os dados inválidos e definir a `:error` propriedade desse widget como uma string explicando o erro.

Próximo:[Definindo novos tipos](#), Anterior:[Dividindo em listas](#), Acima:[Tipos de personalização](#) [Conteúdo][Índice]