

Próximo: [Tipos compostos](#), Acima: [Tipos de personalização](#) [[Conteúdo](#)][[Índice](#)]

15.4.1 Tipos Simples

Esta seção descreve todos os tipos de personalização simples. Para vários desses tipos de personalização, o widget de personalização fornece conclusão em linha com C-M-i ou . M-TAB

sexp

O valor pode ser qualquer objeto Lisp que possa ser impresso e lido. Você pode usar `sexp` como um substituto para qualquer opção, se não quiser perder tempo para descobrir um tipo mais específico a ser usado.

integer

O valor deve ser um número inteiro.

number

O valor deve ser um número (ponto flutuante ou inteiro).

float

O valor deve ser ponto flutuante.

string

O valor deve ser uma string. O buffer de personalização mostra a string sem delimitar `'` caracteres ou `\` citações.

regexp

Como `string`, exceto que a string deve ser uma expressão regular válida.

character

O valor deve ser um código de caractere. Um código de caractere é na verdade um inteiro, mas esse tipo mostra o valor inserindo o caractere no buffer, em vez de mostrar o número.

file

O valor deve ser um nome de arquivo. O widget fornece a conclusão.

(file :must-match t)

O valor deve ser um nome de arquivo para um arquivo existente. O widget fornece a conclusão.

directory

O valor deve ser um diretório. O widget fornece a conclusão.

hook

O valor deve ser uma lista de funções. Esse tipo de personalização é usado para variáveis de gancho. Você pode usar a palavra- `:options` chave em uma variável de gancho `defcustom` para especificar uma lista de funções recomendadas para uso no gancho; Consulte [Definições de Variáveis](#) .

symbol

O valor deve ser um símbolo. Ele aparece no buffer de personalização como o nome do símbolo. O widget fornece a conclusão.

function

O valor deve ser uma expressão lambda ou um nome de função. O widget fornece preenchimento para nomes de funções.

variable

O valor deve ser um nome de variável. O widget fornece a conclusão.

face

O valor deve ser um símbolo que é um nome de rosto. O widget fornece a conclusão.

boolean

O valor é booleano—ou `nil` ou `t`. Observe que usando `choice` e `constjuntos` (consulte a próxima seção), você pode especificar que o valor deve ser `nil` ou `t`, mas também especificar o texto para descrever cada valor de uma maneira que se ajuste ao significado específico da alternativa.

key-sequence

O valor é uma sequência de teclas. O buffer de personalização mostra a sequência de teclas usando a mesma sintaxe da `kbd`função. Consulte [Sequências de teclas](#) .

coding-system

O valor deve ser um nome de sistema de codificação e você pode concluir com `. M-TAB`

color

O valor deve ser um nome de cor válido. O widget fornece preenchimento para nomes de cores, bem como uma amostra e um botão para selecionar um nome de cor de uma lista de nomes de cores mostrados em um `*Cores*` amortecedor.

Próximo: [Tipos compostos](#), Acima: [Tipos de personalização](#) [[Conteúdo](#)][[Índice](#)]