

Próximo:[Aplicando personalizações](#), Anterior:[Definições de Variáveis](#), Acima:[Costumização \[Conteúdo\]](#)
[\[Índice\]](#)

15.4 Tipos de Personalização

Ao definir uma opção de usuário com `defcustom`, você deve especificar seu *tipo de personalização*. Esse é um objeto Lisp que descreve (1) quais valores são legítimos e (2) como exibir o valor no buffer de personalização para edição.

Você especifica o tipo de personalização `defcustom` com a palavra-`:type` chave. O argumento de `:type` é avaliado, mas apenas uma vez quando o `defcustom` é executado, portanto, não é útil que o valor varie. Normalmente usamos uma constante cotada. Por exemplo:

```
(defcustom diff-command "diff"
  "O comando a ser usado para executar diff."
  :type '(string)
  :group 'dif)
```

Em geral, um tipo de personalização é uma lista cujo primeiro elemento é um símbolo, um dos nomes de tipo de personalização definidos nas seções a seguir. Após este símbolo vem uma série de argumentos, dependendo do símbolo. Entre o símbolo de tipo e seus argumentos, você pode opcionalmente escrever pares de valor-palavra-chave (consulte Palavras-[chave de tipo](#)).

Alguns símbolos de tipo não usam argumentos; esses são chamados *tipos simples*. Para um tipo simples, se você não usar nenhum par palavra-chave-valor, poderá omitir os parênteses ao redor do símbolo de tipo. Por exemplo, assim `string` como um tipo de personalização é equivalente a `(string)`.

Todos os tipos de customização são implementados como widgets; veja [Introdução](#) na Biblioteca de Widgets do Emacs, para detalhes.

- [Tipos Simples](#) Tipos de personalização simples: sexo, inteiro, etc.
- [Tipos compostos](#) Crie novos tipos a partir de outros tipos ou dados.
- [Juntando em Listas](#) Junte os elementos na lista com `:inline`.
- [Palavras-chave do tipo](#) Pares de argumento-palavra-chave em um tipo de personalização.
- [Definindo Novos Tipos](#) Dê um nome ao seu tipo.

Próximo:[Aplicando personalizações](#), Anterior:[Definições de Variáveis](#), Acima:[Costumização \[Conteúdo\]](#)
[\[Índice\]](#)