

Próximo:[Tipo de sequência](#), Anterior:[Tipo de personagem](#), Acima:[Tipos de programação](#) [Conteúdo][[Índice](#)]

2.4.4 Tipo de Símbolo

Um *símbolo* no GNU Emacs Lisp é um objeto com um nome. O nome do símbolo serve como a representação impressa do símbolo. No uso comum do Lisp, com um único obarray (consulte [Criando símbolos](#)), o nome de um símbolo é único - não há dois símbolos com o mesmo nome.

Um símbolo pode servir como uma variável, como um nome de função ou para conter uma lista de propriedades. Ou pode servir apenas para ser distinto de todos os outros objetos Lisp, de modo que sua presença em uma estrutura de dados possa ser reconhecida de forma confiável. Em um determinado contexto, geralmente apenas um desses usos é pretendido. Mas você pode usar um símbolo de todas essas maneiras, independentemente.

Um símbolo cujo nome começa com dois pontos (':') é chamado de *símbolo de palavra-chave*. Esses símbolos agem automaticamente como constantes e normalmente são usados apenas para comparar um símbolo desconhecido com algumas alternativas específicas. Consulte [Variáveis Constantes](#).

Um nome de símbolo pode conter qualquer caractere. A maioria dos nomes de símbolos são escritos com letras, dígitos e os caracteres de pontuação '-+/*/'. Tais nomes não requerem pontuação especial; os caracteres do nome são suficientes desde que o nome não se pareça com um número. (Se isso acontecer, escreva um '\' no início do nome para forçar a interpretação como símbolo.) Os caracteres '_~!@\$%^&:<>{}?' são usados com menos frequência, mas também não requerem pontuação especial. Quaisquer outros caracteres podem ser incluídos no nome de um símbolo escapando-os com uma barra invertida. Em contraste com seu uso em strings, no entanto, uma barra invertida no nome de um símbolo simplesmente cita o caractere único que segue a barra invertida. Por exemplo, em uma string, '\t' representa um caractere de tabulação; em nome de um símbolo, no entanto, '\t' apenas cita a carta 't'. Para ter um símbolo com um caractere de tabulação em seu nome, você deve realmente usar uma tabulação (precedida por uma barra invertida). Mas é raro fazer uma coisa dessas.

Nota do Common Lisp: No Common Lisp, as letras minúsculas são sempre dobradas para maiúsculas, a menos que sejam explicitamente escapadas. No Emacs Lisp, letras maiúsculas e minúsculas são distintas.

Aqui estão vários exemplos de nomes de símbolos. Observe que o '+' no quarto exemplo é escapado para evitar que seja lido como um número. Isso não é necessário no sexto exemplo porque o restante do nome o torna inválido como um número.

```
fo; Um símbolo chamado 'foo'.
F00; Um símbolo chamado 'F00', diferente de 'foo'.
1+; Um símbolo chamado '1+'  
      ; (não '+1', que é um número inteiro).
\+1 ; Um símbolo chamado '+1'  
      ; (não é um nome muito legível).
\(*\ 1\ 2\); Um símbolo chamado '(* 1 2)' (um nome pior).
+-*/_~!@$%^&:<>{} ; Um símbolo chamado '+-*/_~!@$%^&:<>{}'.
                      ; Esses caracteres não precisam ser escapados.
```

Como exceção à regra de que o nome de um símbolo serve como sua representação impressa, '##' é a representação impressa de um símbolo interno cujo nome é uma string vazia. Além disso, '#: *fo*' é a representação impressa de um símbolo não interno cujo nome é *foo*. (Normalmente, o leitor Lisp interna todos os símbolos; veja [Criando Símbolos](#).)

Próximo:[Tipo de sequência](#), Anterior:[Tipo de personagem](#), Acima:[Tipos de programação](#) [Conteúdo][[Índice](#)]