

Próximo:[Funções de matriz](#), Anterior:[Funções de sequência](#), Acima:[Vetores de matrizes de sequências](#) [[Conteúdo](#)][[Índice](#)]

6.2 Matrizes

Um objeto *array* possui slots que contêm vários outros objetos Lisp, chamados de elementos do array. Qualquer elemento de um array pode ser acessado em tempo constante. Em contraste, o tempo para acessar um elemento de uma lista é proporcional à posição desse elemento na lista.

O Emacs define quatro tipos de array, todos unidimensionais: *strings* (veja [String Type](#)), *vetores* (veja [Vector Type](#)), *bool-vectors* (veja [Bool-Vector Type](#)) e *char-tables* (veja [Char-Table Type](#)). Vetores e tabelas de caracteres podem conter elementos de qualquer tipo, mas strings podem conter apenas caracteres e vetores bool podem conter apenas `te nil`.

Todos os quatro tipos de array compartilham essas características:

- O primeiro elemento de uma matriz tem índice zero, o segundo elemento tem índice 1 e assim por diante. Isso é chamado de indexação *de origem zero*. Por exemplo, uma matriz de quatro elementos tem índices 0, 1, 2 e 3.
- O comprimento do array é fixo assim que você o cria; você não pode alterar o comprimento de uma matriz existente.
- Para fins de avaliação, a matriz é uma constante, ou seja, ela avalia a si mesma.
- Os elementos de uma matriz podem ser referenciados ou alterados com as funções `arefe` e `aset`, respectivamente (consulte [Funções de matriz](#)).

Ao criar uma matriz, diferente de uma tabela de caracteres, você deve especificar seu comprimento. Você não pode especificar o comprimento de uma tabela de caracteres, porque isso é determinado pelo intervalo de códigos de caracteres.

Em princípio, se você quiser uma matriz de caracteres de texto, poderá usar uma string ou um vetor. Na prática, sempre escolhemos strings para tais aplicações, por quatro motivos:

- Eles ocupam um quarto do espaço de um vetor dos mesmos elementos.
- As strings são impressas de uma maneira que mostra o conteúdo mais claramente como texto.
- Strings podem conter propriedades de texto. Consulte [Propriedades de texto](#).
- Muitos dos recursos especializados de edição e E/S do Emacs aceitam apenas strings. Por exemplo, você não pode inserir um vetor de caracteres em um buffer da mesma forma que insere uma string. Consulte [Strings e Caracteres](#).

Por outro lado, para uma matriz de caracteres de entrada do teclado (como uma sequência de teclas), um vetor pode ser necessário, porque muitos caracteres de entrada do teclado estão fora do intervalo que caberá em uma string. Consulte [Entrada de sequência de teclas](#).

Próximo:[Funções de matriz](#), Anterior:[Funções de sequência](#), Acima:[Vetores de matrizes de sequências](#) [[Conteúdo](#)][[Índice](#)]

