

Próximo:[Modificando listas](#), Anterior:[Listas de construção](#), Acima:[Listas](#) [[Conteúdo](#)][[Índice](#)]

5.5 Modificando as Variáveis da Lista

Essas funções e uma macro fornecem maneiras convenientes de modificar uma lista armazenada em uma variável.

Macro: `push elemento listname`

Esta macro cria uma nova lista cujo CAR é *elemento* e cujo CDR é a lista especificada por *nomedalista*, e salva essa lista em *nomedalista*. No caso mais simples, *listname* é um símbolo sem aspas que nomeia uma lista e essa macro é equivalente a .

```
(setq listname (cons element listname))
```

```
(setq l '(ab))
⇒ (ab)
(pressione 'cl)
⇒ (táxi)
eu
⇒ (táxi)
```

Mais geralmente, *listname* pode ser uma variável generalizada. Nesse caso, essa macro faz o equivalente a . Consulte [Variáveis Generalizadas](#). (setf listname (cons element listname))

Para a popmacro, que remove o primeiro elemento de uma lista, consulte [Elementos da lista](#).

Duas funções modificam listas que são os valores das variáveis.

Função: elemento de símbolo `add-to-list & opcional append compare-fn`

Esta função configura o *símbolo* da variável `consing element` no valor antigo, se o *elemento* ainda não for um membro desse valor. Retorna a lista resultante, atualizada ou não. O valor do *símbolo* deve ser uma lista já antes da chamada. `add-to-list` usa *compare-fn* para comparar o *elemento* com os membros da lista existente; se *compare-fn* for `nil`, ele usará `equal`.

Normalmente, se o *elemento* for adicionado, ele será adicionado à frente do *símbolo*, mas se o argumento opcional *append* for `non-nil`, ele será adicionado no final.

O *símbolo* do argumento não é citado implicitamente; `add-to-list` é uma função comum, como `sete` diferente de `setq`. Cite você mesmo o argumento se é isso que você quer.

Não use esta função quando *símbolo* se referir a uma variável léxica.

Aqui está um cenário mostrando como usar `add-to-list`:

```
(setq foo '(ab))
⇒ (ab)

(adicionar à lista 'foo 'c) ;; Adicionar c.
⇒ (táxi)

(adicionar à lista 'foo 'b) ;; Nenhum efeito.
⇒ (táxi)
```

```
fo;; foofoi alterado.
⇒ (táxi)
```

Uma expressão equivalente para é esta: (add-to-list 'var value)

```
(if ( valor do membro var )
    var
    (setq var ( valor cons var )))
```

Função: elemento de símbolo add-to-ordered-list & order opcional

Esta função configura o *símbolo* da variável inserindo *elemento* no valor antigo, que deve ser uma lista, na posição especificada por *order* . Se o *elemento* já for membro da lista, sua posição na lista é ajustada de acordo com a *ordem* . A associação é testada usando eq. Esta função retorna a lista resultante, atualizada ou não.

A *ordem* geralmente é um número (inteiro ou flutuante) e os elementos da lista são classificados em ordem numérica não decrescente.

ordem também pode ser omitida ou nil. Então a ordem numérica do *elemento* permanece inalterada se já tiver um; caso contrário, o *elemento* não tem ordem numérica. Os elementos sem uma ordem de lista numérica são colocados no final da lista, sem nenhuma ordem específica.

Qualquer outro valor para *ordem* remove a ordem numérica do *elemento* se já tiver uma; caso contrário, é equivalente a nil.

O *símbolo* do argumento não é citado implicitamente; add-to-ordered-list é uma função comum, como set e diferente de setq. Cite você mesmo o argumento, se necessário.

As informações do pedido são armazenadas em uma tabela de hash na propriedade do *símbolo* list-order . *símbolo* não pode se referir a uma variável léxica.

Aqui está um cenário mostrando como usar add-to-ordered-list:

```
(setq foo '())
⇒ nada

(adicionar à lista ordenada 'foo 'a 1) ;; Adicionar a.
⇒ (a)

(adicionar à lista ordenada 'foo 'c 3) ;; Adicionar c.
⇒ (ac)

(adicionar à lista ordenada 'foo 'b 2) ;; Adicionar b.
⇒ (abc)

(adicionar à lista ordenada 'foo 'b 4) ;; Mover b.
⇒ (acb)

(adicionar à lista ordenada 'foo 'd) ;; Anexar d.
⇒ (acd)

(adicionar à lista ordenada 'foo 'e) ;; Adicionare .
⇒ (acbed)

fo;; foofoi alterado.
⇒ (acbed)
```

Próximo:[Modificando listas](#), Anterior:[Listas de construção](#), Acima:[Listas](#) [[Conteúdo](#)][[Índice](#)]