

Próximo:[Lambda Simples](#), Acima:[Expressões lambda](#) [[Conteúdo](#)][[Índice](#)]

### 13.2.1 Componentes de uma expressão lambda

Uma expressão lambda é uma lista que se parece com isso:

```
(lambda ( variáveis-arg ...)
  [ documentação-string ]
  [ declaração interativa ]
  formas do corpo ...)
```

O primeiro elemento de uma expressão lambda é sempre o símbolo `lambda`. Isso indica que a lista representa uma função. A razão pela qual as funções são definidas para começar `lambda` é para que outras listas, destinadas a outros usos, não sejam acidentalmente válidas como funções.

O segundo elemento é uma lista de símbolos — os nomes das variáveis de argumento (consulte [Lista de argumentos](#)). Isso é chamado de *lista lambda*. Quando uma função Lisp é chamada, os valores dos argumentos são comparados com as variáveis na lista lambda, que recebem ligações locais com os valores fornecidos. Consulte [Variáveis Locais](#).

A string de documentação é um objeto string Lisp colocado dentro da definição de função para descrever a função para os recursos de ajuda do Emacs. Consulte [a Documentação da Função](#).

A declaração interativa é uma lista do formulário. Isso declara como fornecer argumentos se a função for usada interativamente. Funções com esta declaração são chamadas de *comandos*; eles podem ser chamados usando ou vinculados a uma chave. As funções que não devem ser chamadas dessa maneira não devem ter declarações interativas. Consulte [Definindo Comandos](#), para saber como escrever uma declaração interativa. (*interactive code-string*)M-x

O resto dos elementos é o *corpo* da função: o código Lisp para fazer o trabalho da função (ou, como diria um programador Lisp, “uma lista de formas Lisp para avaliar”). O valor retornado pela função é o valor retornado pelo último elemento do corpo.

Próximo:[Lambda Simples](#), Acima:[Expressões lambda](#) [[Conteúdo](#)][[Índice](#)]