

Próximo:[Formulários](#), Acima:[Avaliação](#) [Conteúdo][Índice]

10.1 Introdução à Avaliação

O interpretador Lisp, ou avaliador, é a parte do Emacs que calcula o valor de uma expressão que lhe é dada. Quando uma função escrita em Lisp é chamada, o avaliador calcula o valor da função avaliando as expressões no corpo da função. Assim, executar qualquer programa Lisp significa realmente executar o interpretador Lisp.

Um objeto Lisp destinado à avaliação é chamado de *formulário* ou *expressão*⁶. O fato de os formulários serem objetos de dados e não meramente texto é uma das diferenças fundamentais entre linguagens do tipo Lisp e linguagens de programação típicas. Qualquer objeto pode ser avaliado, mas na prática apenas números, símbolos, listas e strings são avaliados com muita frequência.

Nas seções subsequentes, descreveremos os detalhes do que a avaliação significa para cada tipo de formulário.

É muito comum ler um formulário em Lisp e depois avaliar o formulário, mas leitura e avaliação são atividades separadas e podem ser realizadas sozinhas. A leitura em si não avalia nada; ele converte a representação impressa de um objeto Lisp no próprio objeto. Cabe ao chamador de readespecificar se esse objeto é um formulário a ser avaliado ou serve a algum propósito totalmente diferente. Consulte [Funções de entrada](#).

A avaliação é um processo recursivo, e avaliar um formulário geralmente envolve avaliar partes desse formulário. Por exemplo, quando você avalia um formulário de *chamada de função* como (`car x`), o Emacs primeiro avalia o argumento (o subformulário `x`). Após avaliar o argumento, o Emacs *executa* a função (`car`), e se a função for escrita em Lisp, a execução funciona avaliando o *corpo* da função (neste exemplo, no entanto, `car` não é uma função Lisp; é uma função primitiva implementada em C). Consulte [Funções](#), para obter mais informações sobre funções e chamadas de função.

A avaliação ocorre em um contexto chamado *ambiente*, que consiste nos valores e associações atuais de todas as variáveis Lisp (consulte [Variáveis](#)⁷). Sempre que um formulário faz referência a uma variável sem criar uma nova vinculação para ela, a variável avalia o valor fornecido pelo ambiente atual. A avaliação de um formulário também pode alterar temporariamente o ambiente vinculando variáveis (consulte [Variáveis locais](#)).

A avaliação de um formulário também pode fazer alterações que persistem; essas mudanças são chamadas de *efeitos colaterais*. Um exemplo de um formulário que produz um efeito colateral é (`(setq foo 1)`).

Não confunda avaliação com interpretação de tecla de comando. O loop de comando do editor converte a entrada do teclado em um comando (uma função que pode ser chamada interativamente) usando os mapas de teclas ativos e, em seguida, usa `call-interactively` para executar esse comando. A execução do comando geralmente envolve avaliação, se o comando estiver escrito em Lisp; no entanto, esta etapa não é considerada parte da interpretação da tecla de comando. Consulte [Loop de comando](#).

Notas de rodapé

[\(6\)](#)

Às vezes, também é chamado de *expressão S* ou *sexp*, mas geralmente não usamos essa terminologia neste manual.

(7)

Esta definição de “ambiente” especificamente não pretende incluir todos os dados que podem afetar o resultado de um programa.

Próximo:[Formulários](#), Acima:[Avaliação](#) [\[Conteúdo\]](#)[\[Índice\]](#)