

Próximo:[Costumização](#), Anterior:[Funções](#), Acima:[Topo](#) [[Conteúdo](#)][[Índice](#)]

14 Macros

As *macros* permitem que você defina novas construções de controle e outros recursos de linguagem. Uma macro é definida como uma função, mas em vez de dizer como calcular um valor, ela informa como calcular outra expressão Lisp que, por sua vez, calculará o valor. Chamamos essa expressão de *expansão* da macro.

As macros podem fazer isso porque operam nas expressões não avaliadas para os argumentos, não nos valores dos argumentos, como fazem as funções. Eles podem, portanto, construir uma expansão contendo essas expressões de argumento ou partes delas.

Se você estiver usando uma macro para fazer algo que uma função comum poderia fazer, apenas por uma questão de velocidade, considere usar uma função inline. Consulte [Funções embutidas](#) .

- [Macro Simples](#) Um exemplo básico.
- [Expansão](#) Como, quando e por que as macros são expandidas.
- [Compilando Macros](#) Como as macros são expandidas pelo compilador.
- [Definindo Macros](#) Como escrever uma definição de macro.
- [Problemas com Macros](#) Não avalie os argumentos de macro muitas vezes. Não esconda as variáveis do usuário.
- [Macros de recuo](#) Especificando como recuar chamadas de macro.