

Próximo:[Tipo de macro](#), Anterior:[Tipo de tabela de hash](#), Acima:[Tipos de programação](#) [Conteúdo][[Índice](#)]

2.4.13 Tipo de Função

Funções Lisp são códigos executáveis, assim como funções em outras linguagens de programação. Em Lisp, diferentemente da maioria das linguagens, as funções também são objetos Lisp. Uma função não compilada em Lisp é uma expressão lambda: ou seja, uma lista cujo primeiro elemento é o símbolo `lambda`(consulte [Expressões Lambda](#)).

Na maioria das linguagens de programação, é impossível ter uma função sem nome. Em Lisp, uma função não tem nome intrínseco. Uma expressão lambda pode ser chamada como uma função mesmo que não tenha nome; para enfatizar isso, também a chamamos de *função anônima* (consulte [Funções Anônimas](#)). Uma função nomeada em Lisp é apenas um símbolo com uma função válida em sua célula de função (consulte [Definindo Funções](#)).

Na maioria das vezes, as funções são chamadas quando seus nomes são escritos em expressões Lisp em programas Lisp. No entanto, você pode construir ou obter um objeto de função em tempo de execução e chamá-lo com as funções primitivas `funcall` e `apply`. Consulte [Chamando Funções](#) .