

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

2.4.15 Tipo de Função Primitiva

Uma *função primitiva* é uma função que pode ser chamada de Lisp, mas escrita na linguagem de programação C. As funções primitivas também são chamadas *de subrs* ou *funções* internas . (A palavra “subr” é derivada de “sub-rotina”.) A maioria das funções primitivas avaliam todos os seus argumentos quando são chamadas. Uma função primitiva que não avalia todos os seus argumentos é chamada *de forma especial* (consulte [Formas Especiais](#)).

Não importa para o chamador de uma função se a função é primitiva. No entanto, isso importa se você tentar redefinir uma primitiva com uma função escrita em Lisp. A razão é que a função primitiva pode ser chamada diretamente do código C. As chamadas para a função redefinida do Lisp usarão a nova definição, mas as chamadas do código C ainda poderão usar a definição interna. Portanto, **desencorajamos a redefinição de funções primitivas** .

O termo *função* refere-se a todas as funções do Emacs, escritas em Lisp ou C. Consulte [Tipo de Função](#) , para obter informações sobre as funções escritas em Lisp.

As funções primitivas não possuem sintaxe de leitura e imprimem em notação hash com o nome da sub-rotina.

```
(símbolo-função 'carro) ; Acesse a célula de função
                        ; do símbolo.
⇒ #<carro secundário>
(subrp (símbolo-função 'car)); Esta é uma função primitiva?
⇒ t ; sim.
```