

Próximo:[Lista de argumentos](#), Anterior:[Componentes Lambda](#), Acima:[Expressões lambda](#) [[Conteúdo](#)][[Índice](#)]

13.2.2 Um Exemplo Simples de Expressão Lambda

Considere o seguinte exemplo:

```
(lambda (abc) (+ abc))
```

Podemos chamar essa função passando-a para `funcall`, assim:

```
(funcall (lambda (abc) (+ abc))  
 1 2 3)
```

Essa chamada avalia o corpo da expressão `lambda` com a variável `avinculada` a 1, `bvinculada` a 2 e `cvinculada` a 3. A avaliação do corpo soma esses três números, produzindo o resultado 6; portanto, esta chamada para a função retorna o valor 6.

Observe que os argumentos podem ser os resultados de outras chamadas de função, como neste exemplo:

```
(funcall (lambda (abc) (+ abc))  
 1 (* 2 3) (- 5 4))
```

Isso avalia os argumentos 1, `(* 2 3)` e `(- 5 4)` da esquerda para a direita. Em seguida, aplica a expressão `lambda` aos valores de argumento 1, 6 e 1 para produzir o valor 8.

Como esses exemplos mostram, você pode usar um formulário com uma expressão `lambda` como seu CAR para criar variáveis locais e fornecer valores a elas. Nos velhos tempos do Lisp, essa técnica era a única maneira de vincular e inicializar variáveis locais. Mas hoje em dia, é mais claro usar a forma especial `let` para esta finalidade (veja [Variáveis Locais](#)). As expressões `lambda` são usadas principalmente como funções anônimas para passar como argumentos para outras funções (consulte [Funções anônimas](#)) ou armazenadas como definições de funções de símbolo para produzir funções nomeadas (consulte [Nomes de funções](#)).