Próximo:Definindo funções, Anterior:Expressões lambda, Acima:Funções [Conteúdo][Índice]

13.3 Nomeando uma Função

Um símbolo pode servir como o nome de uma função. Isso acontece quando a *célula de função* do símbolo (consulte Componentes do símbolo) contém um objeto de função (por exemplo, uma expressão lambda). Em seguida, o próprio símbolo torna-se uma função válida e que pode ser chamada, equivalente ao objeto de função em sua célula de função.

O conteúdo da célula de função também é chamado de *definição de função* do símbolo . O procedimento de usar a definição de função de um símbolo no lugar do símbolo é chamado de *indireção de função de símbolo* ; veja Função Indireção . Se você não deu a um símbolo uma definição de função, sua célula de função é chamada de *void* e não pode ser usada como uma função.

Na prática, quase todas as funções têm nomes e são referidas por seus nomes. Você pode criar uma função Lisp nomeada definindo uma expressão lambda e colocando-a em uma célula de função (consulte Células de Função). No entanto, é mais comum usar o `defun`formulário especial, descrito na próxima seção. Consulte Definindo Funções .

Damos nomes às funções porque é conveniente referir-se a elas por seus nomes em expressões Lisp. Além disso, uma função Lisp nomeada pode facilmente se referir a si mesma – pode ser recursiva. Além disso, os primitivos só podem ser referidos textualmente por seus nomes, uma vez que os objetos de função primitivos (consulte Tipo de Função Primitivo) não têm sintaxe de leitura.

Uma função não precisa ter um nome exclusivo. Um determinado objeto de função *geralmente* aparece na célula de função de apenas um símbolo, mas isso é apenas uma convenção. É fácil armazená-lo em vários símbolos usando `fset`; então cada um dos símbolos é um nome válido para a mesma função.

Observe que um símbolo usado como nome de função também pode ser usado como variável; esses dois usos de um símbolo são independentes e não entram em conflito. (Este não é o caso em alguns dialetos de Lisp, como Scheme.)

Por convenção, se o símbolo de uma função consiste em dois nomes separados por `'--'`, a função é destinada ao uso interno e a primeira parte nomeia o arquivo que define a função. Por exemplo, uma função nomeada `vc-git--rev-parse`é uma função interna definida em`vc-git.el`. Funções de uso interno escritas em C têm nomes que terminam em `'-interno'`, por exemplo, `bury-buffer-internal`. O código Emacs contribuído antes de 2018 pode seguir outras convenções de nomenclatura de uso interno, que estão sendo eliminadas.

Próximo:Definindo funções, Anterior:Expressões lambda, Acima:Funções [Conteúdo][Índice]