

Próximo:[Símbolos](#), Anterior:[Registros](#), Acima:[Topo](#) [[Conteúdo](#)][[Índice](#)]

8 tabelas de hash

Uma tabela de hash é um tipo muito rápido de tabela de pesquisa, um pouco como uma lista (consulte [Listas de associação](#)) na medida em que mapeia chaves para valores correspondentes. Ele difere de um alist das seguintes maneiras:

- A pesquisa em uma tabela de hash é extremamente rápida para tabelas grandes - na verdade, o tempo necessário é essencialmente *independente* de quantos elementos são armazenados na tabela. Para tabelas menores (algumas dezenas de elementos), as listas ainda podem ser mais rápidas porque as tabelas de hash têm uma sobrecarga mais ou menos constante.
- As correspondências em uma tabela de hash não estão em uma ordem específica.
- Não há como compartilhar estrutura entre duas tabelas de hash, da mesma forma que dois alists podem compartilhar uma cauda comum.

O Emacs Lisp fornece um tipo de dados de tabela de hash de uso geral, juntamente com uma série de funções para operar neles. As tabelas de hash têm uma representação impressa especial, que consiste em '#s' seguido por uma lista especificando as propriedades e o conteúdo da tabela de hash. Consulte [Criando hash](#) . (Notação de hash, a inicial '#' caractere usado nas representações impressas de objetos sem representação de leitura, não tem nada a ver com tabelas de hash. Veja [Representação Impressa](#) .)

Obarrays também são um tipo de tabela de hash, mas são um tipo diferente de objeto e são usados apenas para gravar símbolos internos (consulte [Criando Símbolos](#)).

- [Criando hash](#) Funções para criar tabelas de hash.
- [Acesso de hash](#) Lendo e escrevendo o conteúdo da tabela de hash.
- [Definindo Hash](#) Definição de novos métodos de comparação.
- [Outro Hash](#) Diversos.

Próximo:[Símbolos](#), Anterior:[Registros](#), Acima:[Topo](#) [[Conteúdo](#)][[Índice](#)]