Próximo:Predicados para Strings, Acima:Strings e Personagens [Conteúdo][Índice]

4.1 Fundamentos de Strings e Caracteres

Um caractere é um objeto Lisp que representa um único caractere de texto. No Emacs Lisp, os caracteres são simplesmente inteiros; se um inteiro é um caractere ou não é determinado apenas por como ele é usado. Veja Códigos de Caracteres , para detalhes sobre a representação de caracteres no Emacs.

Uma string é uma sequência fixa de caracteres. É um tipo de sequência chamado *array* , o que significa que seu comprimento é fixo e não pode ser alterado depois de criado (veja Sequences Arrays Vectors). Ao contrário de C, as strings do Emacs Lisp *não* são terminadas por um código de caractere distinto.

Como strings são arrays e, portanto, também sequências, você pode operá-las com as funções gerais de array e sequence documentadas em Sequences Arrays Vectors . Por exemplo, você pode acessar caracteres individuais em uma string usando a função `aref` (consulte Funções de matriz).

Existem duas representações de texto para caracteres não ASCII em strings do Emacs (e em buffers): unibyte e multibyte. Para a maioria das programações em Lisp, você não precisa se preocupar com essas duas representações. Consulte Representações de Texto , para obter detalhes.

Às vezes, as sequências de teclas são representadas como strings de um byte. Quando uma string unibyte é uma sequência de chaves, os elementos string no intervalo de 128 a 255 representam meta caracteres (que são números inteiros grandes) em vez de códigos de caracteres no intervalo de 128 a 255. As strings não podem conter caracteres que tenham os modificadores hyper, super ou alt ; eles podem conter caracteres de controle ASCII , mas nenhum outro caractere de controle. Eles não distinguem maiúsculas e minúsculas em caracteres de controle ASCII . Se você deseja armazenar esses caracteres em uma sequência, como uma sequência de teclas, deve usar um vetor em vez de uma string. Consulte Tipo de caractere , para obter mais informações sobre caracteres de entrada do teclado.

Strings são úteis para conter expressões regulares. Você também pode combinar expressões regulares com strings com `string-match` (consulte Regexp Search). As funções `match-string` (consulte Simple Match Data) e `replace-match`(consulte Substituindo Match) são úteis para decompor e modificar strings depois de corresponder expressões regulares a elas.

Como um buffer, uma string pode conter propriedades de texto para os caracteres nela, bem como os próprios caracteres. Consulte Propriedades de texto . Todas as primitivas Lisp que copiam texto de strings para buffers ou outras strings também copiam as propriedades dos caracteres que estão sendo copiados.

Consulte Text , para obter informações sobre funções que exibem strings ou as copiam em buffers. Consulte Tipo de caractere e Tipo de string para obter informações sobre a sintaxe de caracteres e strings. Consulte Caracteres não ASCII , para funções para converter entre representações de texto e para codificar e decodificar códigos de caracteres. Além disso, observe que *não*`length` deve ser usado para calcular a largura de uma string em exibição; use (consulte Tamanho do texto exibido) em vez disso. `string-width`

Próximo:Predicados para Strings, Acima:Strings e Personagens [Conteúdo][Índice]