

Próximo:[Formatando Strings](#), Anterior:[Comparação de texto](#), Acima:[Strings e Personagens](#) [[Conteúdo](#)][[Índice](#)]

4.6 Conversão de Caracteres e Strings

Esta seção descreve funções para conversão entre caracteres, strings e inteiros. `format`(consulte [Formatando Strings](#)) e `prin1-to-string`(consulte [Funções de Saída](#)) também podem converter objetos Lisp em strings. `read-from-string`(consulte [Funções de entrada](#)) pode converter uma representação de string de um objeto Lisp em um objeto. As funções `string-to-multibyte` e `string-to-unibyte` convertem a representação de texto de uma string (consulte [Convertendo Representações](#)).

Consulte [Documentação](#), para funções que produzem descrições textuais de caracteres de texto e eventos gerais de entrada (`single-key-description` e `text-char-description`). Estes são usados principalmente para fazer mensagens de ajuda.

Função: número de número para string

Esta função retorna uma string que consiste na representação impressa de base dez de *number*. O valor retornado começa com um sinal de menos se o argumento for negativo.

```
(número para string 256)
⇒ "256"
(número para string -23)
⇒ "-23"
(número para string -23,5)
⇒ "-23,5"
```

`int-to-string` é um alias semi-obsoleto para esta função.

Veja também a função `format` em [Formatando Strings](#).

Função: string de string para número e base opcional

Esta função retorna o valor numérico dos caracteres em *string*. Se *base* for diferente `nil`, deve ser um inteiro entre 2 e 16 (inclusive), e os inteiros são convertidos nessa base. Se *a base* for `nil`, então a base dez será usada. A conversão de ponto flutuante só funciona na base dez; não implementamos outras raízes para números de ponto flutuante, porque isso daria muito mais trabalho e não parece útil. Se *string* parece um inteiro, mas seu valor é muito grande para caber em um inteiro Lisp, `string-to-number` retorna um resultado de ponto flutuante.

A análise ignora espaços e tabulações no início de *string*, então lê o máximo de *string* que pode interpretar como um número na base fornecida. (Em alguns sistemas, ela ignora outros espaços em branco no início, não apenas espaços e tabulações.) Se a *string* não puder ser interpretada como um número, esta função retornará 0.

```
(string para número "256")
⇒ 256
(string-to-number "25 é um quadrado perfeito.")
⇒ 25
(string-to-number "X256")
⇒ 0
```

```
(string-to-number "-4.5")  
⇒ -4,5  
(string-to-number "1e5")  
⇒ 100.000,0
```

`string-to-int` é um alias obsoleto para esta função.

Função: *caractere* `char-to-string`

Esta função retorna uma nova string contendo um caractere, *caractere*. Esta função é semi-obsoleta porque a função `string` é mais geral. Consulte [Criando Strings](#).

Função: `string` de *string* para caractere

Esta função retorna o primeiro caractere na *string*. Isso é quase idêntico a `(aref string 0)`, exceto que ele retorna 0 se a string estiver vazia. (O valor também é 0 quando o primeiro caractere da *string* é o caractere nulo, código ASCII 0.) Esta função pode ser eliminada no futuro se não parecer útil o suficiente para ser mantida.

Aqui estão algumas outras funções que podem converter para ou de uma string:

`concat`

Esta função converte um vetor ou uma lista em uma string. Consulte [Criando Strings](#).

`vconcat`

Esta função converte uma string em um vetor. Consulte [Funções vetoriais](#).

`append`

Esta função converte uma string em uma lista. Consulte [Listas de construção](#).

`byte-to-string`

Esta função converte um byte de dados de caractere em uma string unibyte. Consulte [Convertendo Representações](#).

Próximo: [Formatando Strings](#), Anterior: [Comparação de texto](#), Acima: [Strings e Personagens](#) [[Conteúdo](#)][[Índice](#)]