

Próximo:[Sintaxe de Fuga Geral](#), Acima:[Tipo de personagem](#) [[Conteúdo](#)][[Índice](#)]

### 2.4.3.1 Sintaxe de Caracteres Básicos

Como os caracteres são realmente inteiros, a representação impressa de um caractere é um número decimal. Essa também é uma possível sintaxe de leitura para um caractere, mas escrever caracteres dessa maneira em programas Lisp não é uma programação clara. Você deve *sempre* usar os formatos de sintaxe de leitura especiais que o Emacs Lisp fornece para caracteres. Esses formatos de sintaxe começam com um ponto de interrogação.

A sintaxe de leitura usual para caracteres alfanuméricos é um ponto de interrogação seguido pelo caractere; portanto, '?UMA' para o personagem A?'B' para o personagem B, e '?uma' para o personagem a.

Por exemplo:

```
?Q ⇒ 81 ?q ⇒ 113
```

Você pode usar a mesma sintaxe para caracteres de pontuação. No entanto, se o caractere de pontuação tiver um significado sintático especial em Lisp, você deve citá-lo com um '\'. Por exemplo, '?\(' é a maneira de escrever o caractere de parêntese aberto. Da mesma forma, se o personagem for '\', você deve usar um segundo '\' para citar: '?\\'.

Você pode expressar os caracteres control-g, backspace, tab, newline, vertical tab, formfeed, space, return, del e escape como '?\uma', '?\b', '?\t', '?\n', '?\v', '?\f', '?\s', '?\r', '?\d', e '?\e', respectivamente. ('?\s' seguido por um traço tem um significado diferente - ele aplica o modificador Super ao caractere a seguir.) Assim,

```
?\a ⇒ 7 ; controle-g, C-g
?\b ⇒ 8 ; retrocesso, BS , C-h
?\t ⇒ 9 ; tab, TAB , C-i
?\n ⇒ 10 ; nova linha, C-j
?\v ⇒ 11 ; tabulação vertical, C-k
?\f ⇒ 12 ; caractere formfeed, C-l
?\r ⇒ 13 ; retorno de carro, RET , C-m
?\e ⇒ 27 ; caractere de escape, ESC , C-[
?\s ⇒ 32 ; caractere de espaço, SPC
?\\ ⇒ 92 ; caractere de barra invertida, \
?\d ⇒ 127 ;excluir caractere, DEL
```

Essas sequências que começam com barra invertida também são conhecidas como *sequências de escape*, porque a barra invertida desempenha o papel de um caractere de escape; isso não tem nada a ver com o caractere ESC. '\s' destina-se ao uso em constantes de caracteres; em constantes de string, apenas escreva o espaço.

Uma barra invertida é permitida e inofensiva precedendo qualquer caractere sem um significado especial de escape; portanto, '?\+' é equivalente a '?+'. Não há razão para adicionar uma barra invertida antes da maioria dos caracteres. No entanto, você deve adicionar uma barra invertida antes de qualquer um dos caracteres '() [] \; ', e você deve adicionar uma barra invertida antes de qualquer um dos caracteres '| ' ` # . , ' para evitar confundir os comandos do Emacs para editar o código Lisp. Você também deve adicionar uma barra invertida antes dos caracteres Unicode que se assemelham aos ASCII mencionados

anteriormente , para evitar confundir as pessoas lendo seu código. O Emacs destacará alguns caracteres comumente confusos sem escape, como ''' para incentivar isso. Você também pode adicionar uma barra invertida antes de caracteres de espaço em branco, como espaço, tabulação, nova linha e feed de formulário. No entanto, é mais limpo usar uma das sequências de escape facilmente legíveis, como '\t' ou '\s', em vez de um caractere de espaço em branco real, como uma tabulação ou um espaço. (Se você escrever uma barra invertida seguida de um espaço, deverá escrever um espaço extra após a constante de caractere para separá-la do texto a seguir.)

Próximo:[Sintaxe de Fuga Geral](#), Acima:[Tipo de personagem](#) [[Conteúdo](#)][[Índice](#)]