

Anterior:[Criando Buffer-Local](#), Acima:[Variáveis locais de buffer](#) [[Conteúdo](#)][[Índice](#)]

12.11.3 O valor padrão de uma variável local de buffer

O valor global de uma variável com associações de buffer-local também é chamado de valor *padrão*, porque é o valor que está em vigor sempre que nem o buffer atual nem o quadro selecionado têm sua própria associação para a variável.

As funções `default-value` e `setq-default` acessam e alteram o valor padrão de uma variável, independentemente de o buffer atual ter uma associação de buffer-local. Por exemplo, você pode usar `setq-default` para alterar a configuração padrão da `paragraph-start` maioria dos buffers; e isso funcionaria mesmo quando você estiver em um buffer de modo C ou Lisp que tenha um valor local de buffer para essa variável.

Os formulários especiais `defvar` e `defconst` também definem o valor padrão (se eles definirem a variável), em vez de qualquer valor local do buffer.

Função: *símbolo* de valor padrão

Esta função retorna o valor padrão do *símbolo*. Este é o valor que é visto em buffers e frames que não possuem seus próprios valores para esta variável. Se o *símbolo* não for local de buffer, isso é equivalente a `symbol-value` (consulte [Acessando variáveis](#)).

Função: *símbolo* default-boundp

A função `default-boundp` informa se o valor padrão do *símbolo* é nonvoid. Se `(default-boundp 'foo)` retorna `nil`, então `(default-value 'foo)` obterá um erro.

`default-boundp` é para `default-value` como `boundp` é para `symbol-value`.

Formulário especial: `setq-default` [*forma de símbolo*]...

Este formulário especial dá a cada *símbolo* um novo valor padrão, que é o resultado da avaliação do *formulário* correspondente. Ele não avalia o *símbolo*, mas avalia a *forma*. O valor do `setq-default` formulário é o valor do último *formulário*.

Se um *símbolo* não for buffer-local para o buffer atual e não for marcado automaticamente como buffer-local, `setq-default` terá o mesmo efeito que `setq`. Se o *símbolo* for local de buffer para o buffer atual, isso altera o valor que outros buffers verão (desde que não tenham um valor local de buffer), mas não o valor que o buffer atual vê.

```
;; Em buffer 'foo':
(fazer variável local 'buffer-local)
  ⇒ buffer-local
(setq buffer-local 'value-in-foo)
  ⇒ valor-em-foo
(setq-default buffer-local 'novo-padrão)
  ⇒ novo-padrão
buffer-local
  ⇒ valor-em-foo
(valor padrão 'buffer-local)
  ⇒ novo-padrão
```

```
;; No (novo) buffer 'Barra':
buffer-local
  ⇒ novo-padrão
(valor padrão 'buffer-local)
  ⇒ novo-padrão
(setq buffer-local 'outro-padrão)
  ⇒ outro-padrão
(valor padrão 'buffer-local)
  ⇒ outro-padrão

;; De volta ao buffer 'foo':
buffer-local
  ⇒ valor-em-foo
(valor padrão 'buffer-local)
  ⇒ outro-padrão
```

Função: *valor do símbolo padrão definido*

Esta função é como `set-default`, exceto que o *símbolo* é um argumento avaliado comum.

```
(set-default (car '(abc)) 23)
  ⇒ 23
(valor padrão 'a)
  ⇒ 23
```

Uma variável pode ser vinculada (consulte [Variáveis locais](#)) a um valor. Isso torna seu valor global obscurecido pela vinculação; `default-value` retornará o valor dessa ligação, não o valor global, e `set-default` será impedido de definir o valor global (em vez disso, alterará o valor vinculado). As duas funções a seguir permitem referenciar o valor global mesmo se ele estiver sombreado por um `let-binding`.

Função: *símbolo de valor de nível superior padrão*

Essa função retorna o valor padrão *de nível superior de symbol*, que é seu valor fora de qualquer ligação de `let`.

```
(variável defvar 'valor global)
  ⇒ variável
(let ((variável 'let-binding))
  (variável de valor padrão))
  ⇒ vinculação
(let ((variável 'let-binding))
  (variável de valor de nível superior padrão))
  ⇒ valor global
```

Função: *valor do símbolo set-default-toplevel- value*

Esta função define o valor padrão de nível superior do *símbolo para o valor* especificado. Isso é útil quando você deseja definir o valor global de *symbol*, independentemente de seu código ser executado no contexto de `let-binding` do *símbolo*.

Anterior: [Criando Buffer-Local](#), Acima: [Variáveis locais de buffer](#) [[Conteúdo](#)][[Índice](#)]