

Próximo:[Recursos nomeados](#), Anterior:[Carregamento automático](#), Acima:[Carregando](#) [[Conteúdo](#)][[Índice](#)]
]

16.6 Carregamento Repetido

Você pode carregar um determinado arquivo mais de uma vez em uma sessão do Emacs. Por exemplo, depois de reescrever e reinstalar uma definição de função editando-a em um buffer, você pode desejar retornar à versão original; você pode fazer isso recarregando o arquivo de onde veio.

Ao carregar ou recarregar arquivos, lembre-se de que as funções `load` e `load-library` carregam automaticamente um arquivo compilado por byte em vez de um arquivo não compilado de nome semelhante. Se você reescrever um arquivo que pretende salvar e reinstalar, precisará compilar em byte a nova versão; caso contrário, o Emacs carregará o arquivo compilado por byte mais antigo em vez de seu arquivo não compilado mais recente! Se isso acontecer, a mensagem exibida ao carregar o arquivo inclui, '(compilado; nota, a fonte é mais recente)', para lembrá-lo de recompilá-lo.

Ao escrever os formulários em um arquivo de biblioteca Lisp, lembre-se de que o arquivo pode ser carregado mais de uma vez. Por exemplo, pense se cada variável deve ser reinicializada ao recarregar a biblioteca; `defvar` não altera o valor se a variável já estiver inicializada. (Consulte [Definindo Variáveis](#).)

A maneira mais simples de adicionar um elemento a uma lista é assim:

```
(pressione '(leif-mode " Leif") minor-mode-alist)
```

Mas isso adicionaria vários elementos se a biblioteca fosse recarregada. Para evitar o problema, use `add-to-list` (veja [List Variables](#)):

```
(adicionar à lista 'menor-mode-alist' (leif-mode " Leif"))
```

Ocasionalmente, você desejará testar explicitamente se uma biblioteca já foi carregada. Se a biblioteca usa `provide` para fornecer um recurso nomeado, você pode usar `featurep` anteriormente no arquivo para testar se a `provide` chamada foi executada antes (consulte [Recursos nomeados](#)). Alternativamente, você pode usar algo assim:

```
(defvar foo-foi-carregado nil)

(a menos que foo-foi-carregado
  executar somente na primeira vez
  (setq foo-foi-carregado t))
```

Próximo:[Recursos nomeados](#), Anterior:[Carregamento automático](#), Acima:[Carregando](#) [[Conteúdo](#)][[Índice](#)]
]