

GNU Emacs-Referenzkarte

- Version 1.8 -

Im folgenden Text bezeichnet **<C>** eine Kontrolltaste: dazu drücken Sie die **<Control>**- oder **<Strng>**-Taste und gleichzeitig die unmittelbar dahinter angegebene Taste.

Desweiteren bezeichnet **<M>** eine Modifiziertaste: dazu drücken Sie die **<Alternate>**-Taste und gleichzeitig die dahinter angegebene Taste.

Eine Lücke in den angegebenen Zeichenfolgen bedeutet **nicht** die Eingabe eines Leerzeichens (Blank). Wenn ein Leerzeichen eingegeben werden soll, wird dieses ausdrücklich durch **<Space>** dargestellt.

Bei manchen Eingaben folgt noch die Eingabe eines Textstrings (z.B. eines Namens) oder einer Zahl (z.B. einer Zeilennummer); diese Eingaben werden mit **<Return>** beendet bzw. mit **<Escape>** abgebrochen.

Bei einigen Kommandos werden Parameter in einem längeren Dialog abgefragt.

Starten

Um Emacs aufrufen zu können, müssen Sie sein Directory verfügbar haben (z.B. durch einen Eintrag des Pfadnamens in der `.profile`-oder `.login`-Datei).

Emacs aufrufen:

```
emacs
```

Datei mit Namen "hugo" bearbeiten:

```
emacs hugo
```

Beenden

Emacs temporär beenden:

```
<C>z
```

Emacs endgültig verlassen:

```
<C>x <C>c
```

Hilfestellung anfordern

Emacs bietet verschiedene Hilfestellungen in zusätzlichen Fenstern. Generell werden die Hilfen mit `<C>h` aktiviert. Der Anfänger sollte das Tutorial starten. Dies ist auch Voraussetzung zum leichteren Verständnis dieser Referenzkarte.

Start des Tutorials:

```
<C>h t
```

Alle Kommandos zu einem Begriff:

```
<C>h a
```

Funktionsbeschreibungen:

```
<C>h f
```

Funktion auf einer Taste:

```
<C>h c
```

mode-spez. Information:

```
<C>h m
```

Hilfe beenden:

```
<C>x l
```

Hilfe weiterblättern:

```
<Escape> <C>t
```

Fehler beheben

Verlorene Datei wiederherstellen:

```
<M>x file
```

Verlorenen Buffer wiederherstellen:

```
<M>x buffer
```

Undo eines unerwünschten Effekts:

```
<C>x u
```

Bildschirm auffrischen:

```
<C> l
```

Datei-Ein-/Ausgabe

Lesen einer Datei:

```
<C>x <C>f
```

Schreiben einer Datei:

```
<C>x <C>s
```

Einfügen in einer Datei:

```
<C>x l
```

Ersetzen einer Datei:

```
<C>x <C>v
```

Sichern eines Buffers:

```
<C>x <C>w
```

Starten des Dired:

```
<C>x d
```

(Dired ist der "directory editor")

Bewegen in der Textanzeige

Bewegen der Schreibmarke (Cursor)

um die angegebene Einheit:

rückwärts bzw.
vorwärts

Zeichen:

<C>b
<C>f

Wort:

<M>b
<M>f

Zeile:

<C>p
<C>n

Zeilen-Anfang/Ende:

<C>a
<C>e

Satz:

<M>a
<M>e

Paragraph:

<M> [
<M>]

Seite:

<C>x [
<C>x]

sexp:

<C><M>b
<C><M>f

Funktion:

<C><M>a
<C><M>e

Bewegen des Bildschirmausschnitts

nach unten:

<C>v

nach oben:

<M>v

nach links:

<C>x <

nach rechts:

<C>x >

Text suchen

Vorwärts suchen:

<C>s

Rückwärts suchen:

<C>r

Suchen mit regulärem Ausdruck:

<C><M>s

Verlassen des Suchens:

<Escape>

Letztes Zeichen rückgängig:

<Delete>

Laufende Suche abbrechen:

<C>g

Achtung: Fortsetzen des Suchvorganges jeweils durch <C>s bzw. <C>r. Abbrechen einer Suche wirkt sich nur auf die Restarbeiten aus.

Text suchen und ersetzen

Ersetzen starten:

<M>%

Gültige Angaben im Suchen und Ersetzen-Dialog:

Ersetze und mache nächsten Schritt:

<Space>

Ersetze ohne nächsten Schritt:

'

Mache nächsten Schritt ohne Ersetzen:

<Delete>

Ersetze in allen übrigen Fällen:

!

Zurück zum letzten Treffer:

^

Abbrechen

<Escape>

Markieren

der angegebenen Einheit

Paragraph:

<M>h
Seite:
<C>x <C>p
sexp:
<C><M>Q
Funktion:
<C><M>h
Ganzer Buffer:
<C>x h

Marke an dieser Stelle setzen:
<C><Space>
Stelle und Marke austauschen:
<C>x h

Verschieben von Text

Zeichen:
<C>t
Wörter:
<M>t
Zeilen:
<C>x <C>t
Sexp:
<C><M>t

Löschen von Text

Die angegebene Einheit löschen
rückwärts bzw.
vorwärts

Zeichen:
<Delete>
<C>d
Wort:
<M><Delete>
<M>d
Zeile (ab Cursorposition):
<M>o
<C>k <C>k
Satz:
<C>x <Delete>
<M>k

sexp:
<M>- <C><M>k
<C><M>k

Ganze Region löschen:
<C>w

Löschen bis zum nächsten angegebenen Zeichen *char*:
<M>z *char*

Rückholen des zuletzt Gelöschten (yank):
<C>y

Ersetze des zuletzt Zurückgeholten durch das vorher Gelöschte:
<M>y

Mehrere Textfenster verwalten

Splitte Fenster horizontal:
<C>x 5

Splitte Fenster vertikal:
<C>x 2

Springe in nächstes Fenster:
<C>x o

Lösche dieses Fenster:
<C>x 0

Lösche alle anderen Fenster:
<C>x 1

Mache Fenster länger:
<C>x ^

Mache Fenster kürzer:
<M>x window

Mache Fenster schmaler:
<C>x {

Mache Fenster breiter:
<C>x }

Erzeuge ein weiteres Fenster für folgenden Zweck:

für ausgeählten Buffer:
<C>x 4 b

für ausgewählte Datei:
<C>x 4 f

für ein Mail:
<C>x 4 m

für den Dired:

<C>x 4 d
für eine Marken-Suche:
<C>x 4 .

Formatieren des Textes

Neue Zeile hinter aktueller Stelle einsetzen:

<C>o

Zeilenrest eine Zeile nach unten:

<C><M>o

Lösche Leerzeilen um die Stelle:

<C>x <C>o

Lösche Whitespace-Zeichen:

<M>\

Setze exakt ein Blank:

<M><Space>

Fülle Paragraph:

<M>q

Fülle Region:

<M>g

Definiere Füllspalte:

<C>x f

Definiere Prefix am Zeilenanfang:

<C>x .

Einrücken

der angegebenen Einheit

aktuelle Zeile:

<Tab>

Region:

<C><M>\

Sexp:

<C><M>q

Schrifttyp ändern

Ändere

ein Wort oder
eine Region

Alles in Großbuchstaben:

<M>u

<C>x <C>u

Alles in Kleinbuchstaben:

<M>l

<C>x <C>l

Große Anfangsbuchstaben:

<M>c

<M>x *region*

Buffer verwalten

Anderen Buffer auswählen:

<C>x b

Buffer löschen:

<C>x k

Buffer anzeigen:

<C>x <C>b

Zusätzlich kann der Mini-Buffer ediert werden.

Rechtschreibung prüfen

für aktuelles Wort:

<M>\$

für alle Wörter in der Region:

<M>x *region*

für alle Wörter im Buffer:

<M>x *buffer*

Tags (Feldmarken)

Tag finden:

<M>-

Nächstes Vorkommen des Tag

<C>u <M>-

Neuer Tag-file:

<M>x *tag_table*

Setze letzte Suche fort:

<M> ,

Tastenkakros

Beginne Tastenkakro-Definition:

<C>x (

Beende Tastenkakro-Definition:

<C>x)

Führe (letztes) definiertes Tastenkakro aus:

<C>x e

Hänge an letztes Tastenkakro an:

<C>u <C>x (

Benenne letztes Tastenkakro mit *name*:

<M>x *name*

Shell-Kommandos

Ein Shell-Kommando *cmd* ausführen:

<M>! *cmd*

Shell-Kommando *cmd* auf Region anwenden:

<M>| *cmd*

Region als Filter für ein Shellkommando *cmd* erwenden:

<C>u <M>| *cmd*

Starte Shell im Fenster *shell*:

<M>x *shell*

Mailing mit Emacs (Rmail)

Vorwärts blättern:

<Space>

Rückwärts blättern:

<Delete>

Anfang der Nachricht

. (Punkt)

Nächste ungelöschte Nachricht anzeigen (next):

n

Vorige ungelöschte Nachricht anzeigen (previous):

p

Nächste Nachricht:

<M>n

Vorige Nachricht:

<M>p

Lösche aktuelle Nachricht (delete):

d

Lösche Nachricht (mit Sicherung):

<C>d

Hebe Löschung wieder auf:

v

Weiterreichen der Nachricht (forward):

f

Beantworten der Nachricht (reply):

r

Sende Nachricht:

s

Empfange neue Nachrichten (get):

g

Beende Rmail (quit):

q

Zeige Übersicht (Kopfzeilen, headers):

h

Übertrage Nachricht in Rmail-file (output):

o

Info

Zu GNU-Emacs gehört ein Reader (Info_dokument), der in Kapitel gegliedert ist.

Reader öffnen:

<C>h i

Bewegen innerhalb eines Kapitels:

Weiter blättern:

<Space>

Zurück blättern:

<Delete>

Zum Kapitelanfang:

. (Punkt)

Bewegen zwischen den Kapiteln:

Nächstes Kapitel (next):

n

Voriges Kapitel (previous):

p

Nach oben (up):

u

Menu-Auswahl nach Namen:

m

Menu-Auswahl nach Nummer:

number

Querverweis folgen (follow):

f
Rückkehr zu letztem Kapitel (last):
l
Inhaltsverzeichnis anzeigen (directory):
d
Kapitelwahl nach Namen (goto):
>

Sonstiges:

Tutorial starten:
t (tutorial)
Info-Kommandos anzeigen:
?
Info beenden (quit):
q
Kapitel regulär suchen (search):
s

Reguläre Ausdrücke

Die folgenden Zeichen haben innerhalb regulärer Ausdrücke besondere Bedeutung:

0-mal, 1-mal oder mehrfach:
*
1-mal oder mehrfach:
+
0-mal oder 1-mal:
?
Ein einzelnes beliebiges Zeichen:
.
Ein Zeichen aus der Menge *abc*:
[*abc*]
Ein Zeichen nicht in der Menge *abc*:
[!*abc*]
Zeilenanfang:
^
Zeilenende:
\$
Bufferanfang:
\
Bufferende:
\
Wortanfang:
\
<

Wortende:
\
Wortlücke:
\
Weder Wortanfang noch -ende:
\
B

Register

Kopiere Region in das Register:
<C>x x
Füge Registerinhalt an aktueller Stelle ein:
<C>x g
Notiere aktuelle Stelle im Register:
<C>x /
Springe an im Register notierte Stelle (jump):
<C>x j

Programmierung

In Emacs können viele Eigenschaften (z.B. Tastenbelegungen, Kommandos, Variable) mit Hilfe von Emacs-LISP-Programmen individuell maßgeschneidert werden. Lesen Sie dazu bitte die Emacs-Dokumentationen.

Beispiel: Schreiben einer Funktion

```
(defun <command-name> (<arguments>)  
  "<dokumentation>"  
  (interactive "<template>")  
  <body>)
```

Beispiel: Tastenbelegung

```
(global-set-key "\C-cg" 'goto-line)
```

Copyright (C) 1987 Free Software Foundation, Inc.
Entwurf von Stephen Gildes, März 1987 für die GNU Emacs Version 1.8 unter UNIX
Deutsche Version: Günter Schwichtenberg,

Es ist erlaubt, Kopien von dieser Kurzanleitung anzufertigen und weiterzugeben, sofern der Copyright-Vermerk und dieser Weitergabe-Vermerk auf allen Kopien erkennbar bleibt.

*G. Schwichtenberg, Tel.:2346,
Email: g.schwichtenberg@hrz.uni.dortmund.de*