GNU Emacs Referenzkarte

(für Version 19); deutschsprachige Version

`C-k` bedeutet "drücke die Control-Taste und die `k`-Taste gleichzeitig". `M-k` bezieht sich analog auf die Meta-Taste.

Emacs starten und beenden

Um GNU Emacs 19 zu starten, ist einfach `emacs` einzugeben.

Emacs unterbrechen (in X11: ikonifizieren)	`C-z`
Emacs endgültig beenden	`C-x C-c`

Dateien

Einlesen einer Datei in Emacs	`C-x C-f`
Speichern einer Datei auf die Platte	`C-x C-s`
Speichern aller Dateien	`C-x s`
Einfügen des Inhalts der Datei in den Puffer	`C-x i`
Einlesen einer Datei in den aktuellen Puffer	`C-x C-v`
Speichern des Pufferinhalts in eine Datei	`C-x C-w`

Hilfe

Gibt man `C-h` ein, erhält man weitere Anweisungen. Nutzt man erstmals Emacs, sollte man mit `C-h t` das **Tutorial** starten.

Hilfe-Fenster entfernen	`C-x 1`
im Hilfe-Fenster vorwärtsblättern	`ESC C-v`
apropos: Anzeigen der Funktionsnamen, die eine bestimmte Zeichenfolge enthalten	`C-h a`
zur Tastenfolge gehörige Funktion anzeigen	`C-h c`
Beschreibung zu einer Funktion anzeigen	`C-h f`
Modus-spezifische Informationen anzeigen	`C-h m`

Beheben von Fehlern

Abbrechen eines teilweise eingegebenen oder bereits laufenden Befehls	`C-g`
bei Systemabsturz verlorene Datei retten	`M-x recover-file`
Änderung rückgängig machen	`C-x u` or `C-_`
originalen Pufferinhalt wiederherstellen	`M-x revert-buffer`
Bildschirm neu zeichnen	`C-l`

Schrittweises Suchen

Suchen (vorwärts)	`C-s`
Suchen (rückwärts)	`C-r`
Suchen mit regulären Ausdrücken (vorwärts)	`C-M-s`
Suchen mit regulären Ausdr. (rückwärts)	`C-M-r`
vorherige Such-Zeichenfolge auswählen	`M-p`
folgende Such-Zeichenfolge auswählen	`M-n`
schrittweises Suchen beenden	`RET`
letzten Befehl rückgängig machen	`DEL`
aktuelle Suche abbrechen	`C-g`

Mit `C-s` beziehungsweise `C-r` wird die Suche wiederholt. Mit `C-g` wird der noch nicht erledigte Teil abgebrochen.

Bewegung

zu Überspringendes	rückwärts	vorwärts
Zeichen	`C-b`	`C-f`
Wort	`M-b`	`M-f`
Zeile	`C-p`	`C-n`
an Zeilenanfang (oder -ende) gehen	`C-a`	`C-e`
Satz	`M-a`	`M-e`
Absatz	`M-{`	`M-}`
Seite	`C-x [`	`C-x ]`
sexp (syntaktischer Ausdruck)	`C-M-b`	`C-M-f`
Funktion	`C-M-a`	`C-M-e`
an Pufferanfang (oder -ende) gehen	`M-<`	`M->`

zur nächsten Seite blättern	`C-v`
zur vorhergehenden Seite blättern	`M-v`
nach links blättern	`C-x <`
nach rechts blättern	`C-x >`
aktuelle Zeile zur Mitte blättern	`C-u C-l`

Entfernen(kill) und löschen(delete)

zu Entfernendes	rückwärts	vorwärts
Zeichen (nur Löschen)	`DEL`	`C-d`
Wort	`M-DEL`	`M-d`
Zeile (bis zum Ende)	`M-0 C-k`	`C-k`
Satz	`C-x DEL`	`M-k`
sexp	`M-- C-M-k`	`C-M-k`

Region entfernen	`C-w`
kopiere die Region in den kill ring	`M-w`
entfernen bis zum nächsten Zeichen *char*	`M-z` *char*
zuletzt Entferntes wieder einfügen	`C-y`
Wiedereinfügung durch Entferntes ersetzen	`M-y`

Markieren

Marke an aktueller Position setzen	`C-@` or `C-SPC`
Marke und aktuelle Position vertauschen	`C-x C-x`
Marke *arg* Worte weiter setzen	`M-@`
Absatz markieren	`M-h`
Seite markieren	`C-x C-p`
sexp markieren	`C-M-@`
Funktion markieren	`C-M-h`
gesamten Puffer markieren	`C-x h`

Ersetzen mit Nachfrage

interaktives Ersetzen einer Zeichenfolge	`M-%`
dito, mit regulären Ausdrücken	`M-x query-replace-regexp`

Mögliche Antworten im query-replace - Modus sind:

Ersetzen und zum nächsten gehen	`SPC`
Ersetzen und nicht weiterbewegen	`,`
zum nächsten gehen ohne zu ersetzen	`DEL`
alle verbliebenen Sucherfolge ersetzen	`!`
Zurückgehen zum vorherigen Sucherfolg	`^`
den query-replace - Modus beenden	`ESC`
rekursives Editieren (beenden mit `C-M-c`)	`C-r`

Mehrere Fenster (windows)

alle anderen Fenster löschen	`C-x 1`
das aktuelle Fenster löschen	`C-x 0`
Fenster vertikal in zwei Fenster teilen	`C-x 2`
Fenster horizontal in zwei Fenster teilen	`C-x 3`
das andere Fenster vorwärts blättern	`C-M-v`
mit Cursor ins andere Fenster gehen	`C-x o`
Fenster verkleinern (kürzer)	`M-x shrink-window`
Fenster vergrößern (höher)	`C-x ^`
Fenster verkleinern (schmaler)	`C-x {`
Fenster vergrößern (breiter)	`C-x }`
Puffer im anderen Fenster selektieren	`C-x 4 b`
Puffer im anderen Fenster anzeigen	`C-x 4 C-o`
Datei im anderen Fenster öffnen	`C-x 4 f`
Datei im anderen Fenster anzeigen	`C-x 4 r`
Dired im anderen Fenster starten	`C-x 4 d`
Tag-Suche im anderen Fenster starten	`C-x 4 .`

Formatieren

aktuelle Zeile einrücken (modus-abhängig)	`TAB`
Region einrücken (modus-abhängig)	`C-M-\`
sexp einrücken (modus-abhängig)	`C-M-q`
Region fest *arg* Spalten einrücken	`C-x TAB`
nach aktueller Position Newline einfügen	`C-o`
Zeilenrest vertikal abwärts verschieben	`C-M-o`
leere Zeilen um aktuelle Position löschen	`C-x C-o`
Zeile und vorherige verbinden (arg : nächste)	`M-^`
Leerzeichen um aktuelle Position löschen	`M-\`
dito, und ein Leerzeichen einfügen	`M-SPC`
Absatz füllen	`M-q`
letzte Zeile (für Zeilenumbruch) setzen	`C-x f`
Präfix für jede Zeile setzen	`C-x .`

Änderung der Großschreibung

Wort in Großbuchstaben umwandeln	`M-u`
Wort in Kleinbuchstaben umwandeln	`M-l`
dito, aber erster Buchstabe groß	`M-c`
Region in Großbuchstaben umwandeln	`C-x C-u`
Region in Kleinbuchstaben umwandeln	`C-x C-l`
dito, aber erster Buchstabe groß	`M-x capitalize-region`

Der Minipuffer (minibuffer)

Die folgenden Tasten sind im Minipuffer definiert.

so weit wie möglich ergänzen	`TAB`
bis zu einem Wort ergänzen	`SPC`
ergänzenen und ausführen	`RET`
mögliche Ergänzungen anzeigen	`?`
vorherige Minipuffer-Eingabe zurückholen	`M-p`
folgende Minipuffer-Eingabe zurückholen	`M-n`
History-Suche, rückwärts, mit Regexps	`M-r`
History-Suche, vorwärts, mit Regexps	`M-s`
Befehl abbrechen	`C-g`

Mit `C-x ESC ESC` wird der letzte den Minipuffer nutzende Befehl editiert. Folgende Tasten sind definiert:

vorheriger Befehl im Minipuffer	`M-p`
nächster Befehl im Minipuffer	`M-n`

GNU Emacs Referenzkarte

Puffer

anderen Puffer anwählen	C-x b
Liste aller Puffer anzeigen	C-x C-b
Puffer entfernen	C-x k

Vertauschen

Zeichen vertauschen	C-t
Worte vertauschen	M-t
Zeilen vertauschen	C-x C-t
sexps vertauschen	C-M-t

Überprüfung der Rechtschreibung

Überprüfung für aktuelles Wort	M-$
Überprüfung aller Worte einer Region	M-x ispell-region
Überprüfung des gesamten Puffers	M-x ispell-buffer

Tags

ein Tag (eine Definition) finden	M-.
nächstes Auftreten eines Tags finden	C-u M-.
neue Tags-Datei angeben	M-x visit-tags-table
Regexp-Suche in Dateien der Tags-Liste	M-x tags-search
Ersetzen über alle Dateien	M-x tags-query-replace
Suche oder Ersetzen fortsetzen	M-,

Shells

einen Shell-Befehl ausführen	M-!
einen Shell-Befehl auf die Region anwenden	M-|
Region durch einen Shell-Befehl filtern	C-u M-|
Shell im Puffer *shell* starten	M-x shell

Rechtecke

Rechteck in Register kopieren	C-x r r
Rechteck entfernen	C-x r k
Rechteck wieder einfügen	C-x r y
Rechteck erstellen, Text nach rechts schieben	C-x r o
Rechteck mit Leerzeichen überschreiben	M-x clear-rectangle
vor jede Zeile eine Zeichenfolge stellen	M-x string-rectangle

Abkürzungen

globale Abkürzung hinzufügen	C-x a g
modus-lokale Abkürzung hinzufügen	C-x a l
globale Auflösung für Abk. hinzufügen	C-x a i g
modus-lokale Auflösung für Abk. hinzufügen	C-x a i l
Abkürzung explizit auflösen	C-x a e
vorheriges Wort dynamisch auflösen	M-/

Reguläre Ausdrücke (Regexps)

genau ein Zeichen (außer Newline)	. (dot)
null oder mehr Wiederholungen	*
eine oder mehr Wiederholungen	+
null oder eine Wiederholung	?
irgendein Zeichen aus der Menge	[...]
irgendein Zeichen, das nicht in der Menge ist	[^ ...]
Zeilenanfang	^
Zeilenende	$
ein spezielles Zeichen *c* zitieren	*c*
Alternative ("oder")	\\|
Zusammenfassen zu Gruppen	\\(... \\)
*n*te Gruppe	*n*
Pufferanfang	\\\`
Pufferende	\\'
Wortgrenze	\b
weder Wortanfang noch Wortende	\B
Wortbeginn	\\<
Wortende	\\>
Zeichen, das in einem Wort vorkommen kann	\w
Zeichen, das nicht in \w enthalten ist	\W
Zeichen mit Syntax *c*	\s*c*
Zeichen, das nicht die Syntax *c* hat	\S*c*

Register

Region in Register speichern	C-x r s
Registerinhalt in Puffer einfügen	C-x r i
aktuelle Position im Register speichern	C-x r SPC
zur im Register gespeicherten Position gehen	C-x r j

Info

Info-Dokumentationssystem starten	C-h i
Bewegen innerhalb eines Knotens:	
vorwärts blättern	SPC
rückwärts blättern	DEL
zum Anfang des Knotens gehen	. (dot)
Bewegen zwischen Knoten:	
nächster Knoten	n
vorhergehender Knoten	p
nach oben gehen	u
Menü-Eintrag über Namen anwählen	m
*n*ten Menü-Eintrag (*n*=1–5) anwählen	*n*
dem Querverweis folgen (zurück mit l)	f
zurück zum letzten Knoten gehen	l
zurück zum Verzeichnis-Knoten gehen	d
beliebigen Knoten über Namen anwählen	g
Weiteres:	
Info-Tutorial starten	h
Liste der Info-Befehle anzeigen	?
Info beenden	q
im Knoten nach Regexp suchen	s

Tastatur-Makros

Definieren eines Makros beginnen	C-x (
Definieren eines Makros beenden	C-x)
zuletzt definiertes Makro ausführen	C-x e
an zuletzt definiertes Makro anhängen	C-u C-x (
zuletzt definiertes Makro benennen	M-x name-last-kbd-macro
Lisp-Definition in Puffer einfügen	M-x insert-kbd-macro

Arbeit mit Emacs Lisp

vor akt. Postition stehende sexp auswerten	C-x C-e
aktuelles defun auswerten	C-M-x
aktuelle Region auswerten	M-x eval-region
gesamten Puffer auswerten	M-x eval-current-buffer
in Minipuffer einlesen und auswerten	M-ESC
letzten Minipuffer-Befehl nochmals auswerten	C-x ESC ESC
Emacs-Lisp-Datei lesen und auswerten	M-x load-file
Laden aus den System-Verzeichnissen	M-x load-library

Einfache Anpassungen

Es folgen Beispiele zum Binden globaler Tastenfolgen. Statt "\M-#" ist stets "\e#" zu verwenden.

```
(global-set-key "\C-cg" 'goto-line)
(global-set-key "\C-x\C-k" 'kill-region)
(global-set-key "\e#" 'query-replace-regexp)
```

Ein Beispiel zum Setzen einer Variable in Emacs Lisp:

```
(setq backup-by-copying-when-linked t)
```

Befehle schreiben

```
(defun command-name (args)
  "documentation"
  (interactive "template")
  body)
```

Ein Beispiel:

```
(defun this-line-to-top-of-window (line)
  "Reposition line point is on to top of window.
With ARG, put point on line ARG.
Negative counts from bottom."
  (interactive "P")
  (recenter (if (null line)
                0
              (prefix-numeric-value line))))
```

Das Argument zu `interactive` ist eine Zeichenfolge, die angibt, wie die Argumente bei interaktiver Nutzung angefordert werden sollen. Mit `C-h f interactive` erhält man weitere Informationen.