GNU Emacs Reference Card

(for version 18)

Starting Emacs

To read in a file to edit, see Files, below. To enter Emacs, just type its name: emacs

Leaving Emacs

exit Emacs permanently suspend Emacs (the usual way of leaving it) C-z C-x C-c

Files

run Dired, the directory editor	write buffer to a specified file	replace this file with the file you really want	insert contents of another file into this buffer	save a file back to disk	read a file into Emacs
C-x d	C-x	C- x	C-x i	C- x	C-x
Ф	C-x C-W	C-x C-v	μ.	C-x C-s	C-x C-f

Getting Help

card assumes you know the tutorial.) If you are a first-time user, type C-h t for a tutorial. (This The Help system is simple. Type C-h and follow the directions.

get rid of Help window scroll Help window	C-x 1 ESC C-v
apropos: show commands matching a string C-h a	C-h a
show the function a key runs	C-h c
describe a function	C-h f
get mode-specific information	C-h m

Error Recovery

abort partially typed or executing command C-g redraw garbaged screen restore a buffer to its original contents undo an unwanted change recover a file lost by a system crash M-x recover-file M-x revert-buffer C-1 C-x u or C-_

Incremental Search

abort current search	undo effect of last character	exit incremental search	Use $\mathtt{C-s}$ or $\mathtt{C-r}$ again to repeat the search in either direction.	regular expression search	search backward	search forward
C-g	DEL	ESC	ne search in either direction.	C-M-s	C-r	C-s

not done, otherwise it aborts the entire search. If Emacs is still searching, C-g will cancel the part of the search

© 1987 Free Software Foundation, Inc. Permissions on back. v1.9

Motion

	character	entity to move over	Cursor motion:
	С-Ъ	backward	
4	Ċ	fc	

entity to move over	backward	forward
character	C-b	C-f
word	M-P	M-f
line	с-р	C-n
go to line beginning (or end)	C-a	С-е
sentence	M-a	М-е
paragraph	M-[M-]
page	C-x [C-x]
sexp	C-M-b	C-M-f
function	C-M-a	C-M-e
go to buffer beginning (or end)	M-<	M->
Screen motion:		

Killing and Deleting

scroll right scroll left

scroll to next screen scroll to previous screen

yank back last thing killed replace last yank with previous kill	kill region kill to next occurrence of char	entity to kill character (delete, not kill) word line (to end of) sentence sexp
С-у М-у	C-#	backward DEL M-DEL M-0 C-k C-x DEL M C-M-k
7 7	C-w M-z char	forward C-d M-d C-k M-k C-M-k

Marking

set mark here	C-@ or C-SPC
exchange point and mark	C-x C-x
set mark arg words away	M-0
mark paragraph	M-h
mark page	с-х с-р
mark sexp	C-M-0
mark function	C-M-h
mark entire buffer	C-x h

Query Replace

enter recursive edit (C-M-c to exit)	exit query-replace	back up to the previous match	replace all remaining matches	skip to next without replacing	replace this one, don't move	replace this one, go on to next	Valid responses in query-replace mode are	using regular expressions	interactively replace a text string
C-r	ESC	,		DEL	•	SPC	de are	M-x query-replace-regexp	M-%

Multiple Windows

ind tag in other window	run Dired in other window	compose mail in other window	find file in other window	select a buffer in other window	grow window wider	shrink window narrower	grow window taller	shrink window shorter	switch cursor to another window	scroll other window	split window in 2 horizontally	split window in 2 vertically	delete this window	delete all other windows
C=X 4 .	C-x 4 d	C-x 4 m	C-x 4 f	C-x 4 b	C-x }	C-x {	C-x ^	M-x shrink-window	C-x o	C-M-v	C-x 5	C-x 2	C-x 0	C= X 1

Formatting

C-x < C-ν M-ν

C-x >

fill paragraph fill region set fill column set prefix each line starts with	insert newline after point move rest of line vertically down delete blank lines around point delete all whitespace around point put exactly one space at point	indent current line (mode-dependent) indent region (mode-dependent) indent sexp (mode-dependent) indent region rigidly arg columns
M-9 C-x f	C-o C-M-o C-x C-o M-\	TAB C-M-\ C-M-q C-x TAB

Case Change

capitalize region	lowercase region	uppercase region	capitalize word	lowercase word	uppercase word	
M-x capitalize-region	C-x C-1	C-x C-u	M -c	M-T	M-u	

The Minibuffer

The following keys are defined in the minibuffer.

abort command	show possible completions	complete and execute	complete up to one word	complete as much as possible	
C-86	?	RET	SPC	TAB	

the minibuffer. The following keys are then defined. Type C-x ESC to edit and repeat the last command that used

next minibuffer command	previous minibuffer command
M-n	M-p

GNU Emacs Reference Card

Buffers

kill a buffer	list all buffers	select another buffer
C-x k	C-x C-b	C-x b

Transposing

transpose sexps	transpose lines	transpose words	transpose characters
C-M-t	C-x C-t	M – t	C-t

Spelling Check

check spelling of entire buffer	check spelling of all words in region	check spelling of current word
M-x spell-buffer	M-x spell-region	M-\$

Tags

continue last tags search or query-replace	query replace on all the files	regexp search on all files in tags table	specify a new tags file	find next occurrence of tag	find tag
place M-,	M-x tags-query-replace	e M-x tags-search	M-x visit-tags-table	C-u M	M

Shells

start a shell in window *shell*	filter region through a shell command	run a shell command on the region	execute a shell command
M-x shell	C-u M-	M-	i – M

Rmail

show summary of headers	output message in Unix-mail style	output message to another Rmail file	quit Rmail	get newly arrived mail	send mail	forward message to someone	reply to message	undelete message	delete message and back up	delete message	previous message	next message	previous non-deleted message	next non-deleted message	beginning of message	scroll reverse	scroll forward
h	C-0	0	д	019	m	Ħ	Ħ	п	C-d	a.	M-b	M-n	P	n	. (dot)	DEL	SPC

Regular Expressions

The following have special meaning inside a regular expression.

\S c	character with syntax not c
\sc	character with syntax c
/ 15	any non-word-syntax character
/H	any word-syntax character
V	end of word
<u>`</u>	beginning of word
\B	not beginning or end of word
\b	word break
/,	end of buffer
′,	beginning of buffer
\n	nth group
\(\lambda \cdots \cdot\)	grouping
_	alternative ("or")
10	quote a special character c
€9	end of line
,	beginning of line
「 ` : _	any character not in set
[]	any character in set
٠,	zero or one repeat
+	one or more repeats
*	zero or more repeats
. (dot)	any single character

Registers

move point to saved location	save point in register	insert register contents	copy region to register
C- x j	C-x /	C-x g	C-x x

Info

enter the Info documentation reader	C-h i
Moving within a node:	
scroll forward	SPC
scroll reverse	DEL
beginning of node	. (dot)
Moving between nodes:	
next node	n
previous node	P
move up	п
select menu item by name	m
select n th menu item by number $(1-5)$	n
follow cross reference (return with 1)	Ħ
return to last node you saw	L
return to directory node	۵
go to any node by name	09
Other:	
run Info tu torial	h
list Info commands	٠.
quit Info	Ф
search nodes for regexp	to.

Keyboard Macros

insert lisp definition in buffer	name last keyboard macro	append to last keyboard macro	execute last-defined keyboard macro	end keyboard macro definition	start defining a keyboard macro	
M-x insert-kbd-macro	M-x name-last-kbd-macro	C-u C-x (cro C-x e	C-x)	C-x (

Commands Dealing with Emacs Lisp

M-x load-library	load from standard system directory
M-x load-file	read and eval Emacs Lisp file
C-x ESC	re-execute last minibuffer command
M-ESC	read and eval minibuffer
M-x eval-current-buffer	eval entire buffer M-x
M-x eval-region	eval region
C-M-x	eval current defun
C-X C-e	eval sexp before point

Simple Customization

Note that you cannot say " $\mbox{M-#"}$; you must say " $\ensuremath{\text{e\#"}}$. Here are some examples of binding global keys in Emacs Lisp.

```
(global-set-key "\G-cg" 'goto-line)
(global-set-key "\e\G-r" 'isearch-backward-regexp)
(global-set-key "\e#" 'query-replace-regexp)
```

An example of setting a variable in Emacs Lisp:

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

Writing Commands

```
Negative counts from bottom."
                                                                                the screen. With ARG, put point on line ARG
                                                                                                                                            (defun this-line-to-top-of-screen (line)
                                                                                                                                                                                An example:
                                                                                                                                                                                                                                                                                                          (defun (command-name) ((args))
                                                                                                               "Reposition line point is on to the top of
                                                                                                                                                                                                                                                (interactive "(template)")
                                                                                                                                                                                                                                                                               "\langle documentation \rangle"
(recenter (if (null line)
                          (interactive "P")
                                                                                                                                                                                                                          (body))
```

C-h f interactive for more information. the arguments when the function is called interactively. Type The argument to interactive is a string specifying how to get

(prefix-numeric-value line))))

Copyright © 1987 Free Software Foundation, Inc designed by Stephen Gildea, March 1987 v1.9 for GNU Emacs version 18 on Unix systems

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all

dation, Inc., 675 Massachusetts Ave, Cambridge MA 02139 For copies of the GNU Emacs manual, write to the Free Software Foun