

GNU Emacs Reference Card

(for version 21)

Starting Emacs

To enter GNU Emacs 21, just type its name: emacs

To read in a file to edit, see Files, below.

Leaving Emacs

suspend Emacs (or iconify it under X)	C-z
exit Emacs permanently	C-x C-c

Files

read a file into Emacs	C-x C-f
save a file back to disk	C-x C-s
save all files	C-x s
insert contents of another file into this buffer	C-x i
replace this file with the file you really want	C-x C-v
write buffer to a specified file	C-x C-w
version control checkin/checkout	C-x C-q

Getting Help

The help system is simple. Type C-h (or F1) and follow the directions. If you are a first-time user, type C-h t for a tutorial.

remove help window	C-x l
scroll help window	C-M-v
apropos: show commands matching a string	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
recover a file lost by a system crash	M-x recover-file
undo an unwanted change	C-x u or C-_
restore a buffer to its original contents	M-x revert-buffer
redraw garbaged screen	C-l

Incremental Search

search forward	C-s
search backward	C-r
regular expression search	C-M-s
reverse regular expression search	C-M-r
select previous search string	M-p
select next later search string	M-n
exit incremental search	RET
undo effect of last character	DEL
abort current search	C-g

Use C-s or C-r again to repeat the search in either direction. If Emacs is still searching, C-g cancels only the part not done.

Misc
Redo last complex command
F12 (LSCE)

A faire avant
cp ~jypeter/.emacs ~
cp ~jypeter/.abbrev_defs ~

or F1 (LSCE)

Motion

entity to move over
character
word
line
● go to line beginning (or end)
sentence
paragraph
page
sexp
function
● go to buffer beginning (or end)
scroll to next screen
scroll to previous screen
scroll left
scroll right
scroll current line to center of screen
go to line number
Killing and Deleting

backward	forward
C-b	C-f
M-b	M-f
C-p	C-n
C-a	C-e
M-a	M-e
M-{	M-}
C-x [C-x]
C-M-b	C-M-f
C-M-a	C-M-e
M-<	M->
	C-v
	M-v
	C-x <
	C-x >
	C-u C-l
	C-x C-g

or C-↑ C-↓ (LSCE)
or Page Up Down
or Page Up

entity to kill
character (delete, not kill)
word
● line (to end of)
sentence
sexp

backward	forward
DEL	C-d
M-DEL	M-d
M-O C-k	● C-k
C-x DEL	M-k
M-- C-M-k	C-M-k

● kill **region**
copy region to kill ring
kill through next occurrence of *char*
● yank back last thing killed
replace last yank with previous kill

C-w or double-click on right button
M-w
M-z *char*
C-y or click on middle button
M-y

Marking

set mark here
exchange point and mark
set mark *arg* words away
mark **paragraph**
mark **page**
mark **sexp**
mark **function**
mark entire **buffer**

C-@ or C-SPC
C-x C-x
M-@
M-h
C-x C-p
C-M-@
C-M-h
C-x h

Query Replace

interactively replace a text string
● using regular expressions M-x query-replace-regexp
Valid responses in query-replace mode are

● replace this one, go on to next	SPC
replace this one, don't move	,
● skip to next without replacing	DEL
● replace all remaining matches	!
back up to the previous match	~
● exit query-replace	RET
enter recursive edit (C-M-c to exit)	C-r

or F2 (LSCE)

Multiple Windows

When two commands are shown, the second is for “other frame.”

delete all other windows	C-x 1
● split window, above and below	● C-x 2 C-x 5 2
● delete this window	● C-x 0 C-x 5 0
split window, side by side	C-x 3
scroll other window	C-M-v
switch cursor to another window	C-x o C-x 5 o
select buffer in other window	C-x 4 b C-x 5 b
display buffer in other window	C-x 4 C-o C-x 5 C-o
find file in other window	C-x 4 f C-x 5 f
find file read-only in other window	C-x 4 r C-x 5 r
run Dired in other window	C-x 4 d C-x 5 d
find tag in other window	C-x 4 . C-x 5 .
grow window taller	C-x ^
shrink window narrower	C-x {
grow window wider	C-x }

Formatting

indent current line (mode-dependent)	TAB
indent region (mode-dependent)	C-M-\
indent sexp (mode-dependent)	C-M-q
indent region rigidly <i>arg</i> columns	C-x TAB
insert newline after point	C-o
move rest of line vertically down	C-M-o
delete blank lines around point	C-x C-o
join line with previous (with <i>arg</i> , next)	M-^
delete all white space around point	M-\
put exactly one space at point	M-SPC
● fill paragraph	M-q
set fill column	C-x f
set prefix each line starts with	C-x .
set face	M-g

Case Change

● uppercase word	M-u
● lowercase word	M-l
● capitalize word	M-c
uppercase region	C-x C-u
lowercase region	C-x C-l

The Minibuffer

The following keys are defined in the minibuffer.

● complete as much as possible	TAB
● complete up to one word	SPC
complete and execute	RET
show possible completions	?
fetch previous minibuffer input	M-p
fetch later minibuffer input or default	M-n
regexp search backward through history	M-r
regexp search forward through history	M-s
abort command	C-g

Type C-x ESC ESC to edit and repeat the last command that used the minibuffer. Type F10 to activate the menu bar using the minibuffer.

GNU Emacs Reference Card

Buffers

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

Transposing

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

Spelling Check

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

Tags

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

Shells

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

Rectangles

copy rectangle to register	C-x r r
kill rectangle	C-x r k
yank rectangle	C-x r y
open rectangle, shifting text right	C-x r o
blank out rectangle	C-x r c
prefix each line with a string	C-x r t

Abbrevs

add global abbrev	C-x a g
add mode-local abbrev	C-x a l
add global expansion for this abbrev	C-x a i g
add mode-local expansion for this abbrev	C-x a i l
explicitly expand abbrev	C-x a e
expand previous word dynamically	M-/

Regular Expressions

any single character except a newline	.	(dot)
zero or more repeats	*	
one or more repeats	+	
zero or one repeat	?	
quote regular expression special character <i>c</i>	\	<i>c</i>
alternative ("or")		
grouping	(...)
same text as <i>n</i> th group	\	<i>n</i>
at word break	\	b
not at word break	\	B

entity	match start	match end
line	^	\$
word	\<	\>
buffer	\'	\'
class of characters	match these	match others
explicit set	[...]	[^ ...]
word-syntax character	\w	\W
character with syntax <i>c</i>	\s <i>c</i>	\S <i>c</i>

International Character Sets

specify principal language	M-x	set-language-environment
show all input methods	M-x	list-input-methods
enable or disable input method	C-\	
set coding system for next command	C-x	RET <i>c</i>
show all coding systems	M-x	list-coding-systems
choose preferred coding system	M-x	prefer-coding-system

Info

enter the Info documentation reader	C-h	i
find specified function or variable in Info	C-h	C-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	u
select menu item by name	m
select <i>n</i> th menu item by number (1-9)	<i>n</i>
follow cross reference (return with 1)	f
return to last node you saw	l
return to directory node	d
go to any node by name	g

Other:

run Info tutorial	h
quit Info	q
search nodes for regexp	M-s

Registers

save region in register	C-x r s
insert register contents into buffer	C-x r i
save value of point in register	C-x r SPC
jump to point saved in register	C-x r j

Keyboard Macros

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

Commands Dealing with Emacs Lisp

eval sexp before point	C-x C-e
eval current defun	C-M-x
eval region	M-x eval-region
read and eval minibuffer	M-:
load from standard system directory	M-x load-library

Simple Customization

customize variables and faces M-x customize

Making global key bindings in Emacs Lisp (examples):

```
(global-set-key "\C-cg" 'goto-line)
(global-set-key "\M-#" 'query-replace-regexp)
```

Writing Commands

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

An example:

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

The interactive spec says how to read arguments interactively.
Type C-h f *interactive* for more details.

Copyright © 1997 Free Software Foundation, Inc.
v2.2 for GNU Emacs version 21, 1997
designed by Stephen Gildea

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

For copies of the GNU Emacs manual, write to the Free Software Foundation, Inc.,
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Dired Reference Card

(based on Dired in GNU Emacs 21.x)

Commands marked with **(DX)** require dired-x

Entering and Exiting Dired

run dired	C-x d
dired the directory of the file you are editing	C-x C-j (DX)
quit dired	q

Motion Commands

move up to previous line	p
move down to next line	n
move up to previous directory line	<
move down to next directory line	>
move to next marked file	M-}
move to previous marked file	M-{
move up to previous subdirectory	M-C-p
move down to next subdirectory	M-C-n
move to parent directory	-
move to first child subdirectory	M-C-d

Mouse Commands

visit file	Mouse_Button_2
popup menu	Control-Mouse_Button_3

Immediate Actions on Files

visit current file	f
view current file	v
visit current file in other window	o
visit current file in other frame	w
display current file	C-u o
create a new subdirectory	+
compare file at point with the one at mark	=

Marking and Unmarking Files

mark a file or subdirectory for later commands	m
unmark a file or all files of a subdirectory	u
unmark all marked files in a buffer	M-delete
mark files with a given extension	* .
mark all directories	* /
mark all symlinks	* @
mark all executables	* *
invert marking	* t
mark all files in the current subdir	* s
mark file names matching a regular expression	* %
change the marks to a different character	* c
mark files for which Elisp expression returns t	* ((DX)

Getting Help

dired help	h
dired summary (short help) and error log	?

Commands on Files Marked or Specified by the Prefix

copy file(s)	C
rename a file or move files to another directory	R
change ownership of file(s)	O
change the group of the file(s)	G
change mode of file(s)	M
print file(s)	P
convert filename(s) to lower case	% l
convert filename(s) to upper case	% u
delete marked (as opposed to flagged) files	X
uencode or udecode file(s)	U
compress or uncompress file(s)	Z
run info on file	I (DX)
make symbolic link(s)	S
make relative symbolic links	Y
make hard link(s)	H
search files for a regular expression	A
query replace regular expression	Q
byte-compile files	B
load files	L
shell command on file(s)	!

Flagging Files for Deletion

Unmark commands remove delete flags

flag file for deletion	d
backup and remove deletion flag	delete
flag all backup files (file names ending in ~)	~
flag all auto-save files	#
flag various intermediate files	&
flag numeric backups (ending in .~1~, .~2~, etc.)	.
execute the deletions requested (flagged files)	x
flag files matching a regular expression	% d

Modifying the Dired Buffer

insert a subdirectory into this buffer	i
remove marked files from the listing	k
remove a subdir listing	C-u k
re-read all directories (retains all marks)	g
toggle sorting of current subdir by name/date	s
edit ls switches	C-u s
recover marks, hidden lines, and such	C-_
hide all subdirectories	M-\$
hide or unhide subdirectory	\$

Regular Expression Commands

mark filenames matching a regular expression	% m
copy marked files by regexp	% C
rename marked files by regexp	% R
hardlink	% H
symlink	% S
symlink, with relative paths	% Y
mark for deletion	% d

Dired and Find

dired file(s) whose name matches a pattern	M-x find-name-dired
dired file(s) that contain pattern	M-x find-grep-dired
dired file(s) based on find output	M-x find-dired