

# XEmacs Reference Card

(for version 21.0+)

## Starting Emacs

To enter XEmacs, just type its name: `xemacs`

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

## Leaving Emacs

suspend Emacs (or iconify frame under X)

exit Emacs permanently

C-z C-c

## Files

read a file into Emacs

save a file back to disk

save all files

insert contents of another file into this buffer

replace this file with the file you really want

write buffer to a specified file

C-x C-f  
C-x C-s  
C-x s  
C-x i  
C-x C-v  
C-x C-w

## Getting Help

The Help system is simple. Type C-h and follow the directions.

If you are a first-time user, type C-h t for a tutorial.

quit Help window

q space

scroll Help window

apropos: show commands matching a string

show the function a key runs

describe a function

C-h a  
C-h c  
C-h f  
C-h m

## Error Recovery

abort partially typed or executing command

recover a file lost by a system crash

recover files from a previous Emacs session

undo an unwanted change

restore a buffer to its original contents

redraw garbagged screen

C-g  
M-x recover-file  
M-x recover-session  
C-x u or C-  
M-x revert-buffer  
C-1

## Incremental Search

search forward

search backward

regular expression search

reverse regular expression search

select previous search string

select next later search string

exit incremental search

undo effect of last character

abort current search

C-s  
C-r  
C-M-s  
C-M-r  
M-p  
M-n  
RET  
DEL  
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.

## 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

backward forward  
C-b C-f  
M-b M-f  
C-p C-n  
C-a C-e  
M-  
C-x [ C-x ]  
C-M-b C-M-f  
C-M-a C-M-e  
M-< M->

## Killing and Deleting

entity to kill

character (delete, not kill)

word

line (to end of)

sentence

sexp

kill region

copy region to kill ring

kill through next occurrence of char

yank back last thing killed

replace last yank with previous kill

backward forward  
DEL C-d  
M-DEL M-d  
M-0 C-k C-k  
C-x DEL M-k  
M- C-M-k C-M-k

## Marking

set mark here

exchange point and mark

set mark any 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

Valid responses in query-replace mode are

replace this one, go on to next

replace this one, don't move

skip to next without replacing

replace all remaining matches

back up to the previous match

exit query-replace

enter recursive edit (C-M-c to exit)

delete match and enter recursive edit

M-%  
M-x query-replace-regexp  
SPC or y  
,  
DEL or n  
!  
~  
ESC  
C-r  
C-w

## Multiple Windows

delete all other windows

delete this window

split window in two vertically

split window in two horizontally

scroll other window

switch cursor to another window

shrink window shorter

grow window taller

shrink window narrower

grow window wider

select buffer in other window

display buffer in other window

find file in other window

find file read-only in other window

run Dired in other window

find tag in other window

C-x 1  
C-x 0  
C-x 2  
C-x 3  
C-M-v  
C-x o  
M-x shrink-window  
C-x ~  
C-x {  
C-x }  
C-x j  
C-x 4 b  
C-x 4 C-o  
C-x 4 f  
C-x 4 r  
C-x 4 d  
C-x 4 .

## Formatting

indent current line (mode-dependent)

indent region (mode-dependent)

indent sexp (mode-dependent)

indent region rigidly any columns

insert newline after point

move rest of line vertically down

delete blank lines around point

join line with previous (with arg, next)

delete all white space around point

put exactly one space at point

fill paragraph

set fill column

set prefix each line starts with

TAB  
C-M-\  
C-M-q  
C-x TAB  
C-o  
C-M-o  
C-x C-o  
M-\  
M-SPC  
M-q  
C-x f  
C-x .

## Case Change

uppercase word

lowercase word

capitalize word

uppercase region

lowercase region

capitalize region

M-u  
M-l  
M-c  
C-x C-u  
C-x C-l  
M-x capitalize-region

## The Minibuffer

The following keys are defined in the minibuffer.

complete as much as possible

complete up to one word

complete and execute

show possible completions

fetch previous minibuffer input

fetch next later minibuffer input

reexp search backward through history

reexp search forward through history

abort command

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

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

# XEmacs Reference Card

## Buffers

select another buffer  
 list all buffers  
 kill a buffer

C-x b  
 C-x C-b  
 C-x k

## Transposing

transpose characters  
 transpose words  
 transpose lines  
 transpose sexprs

C-t  
 M-t  
 C-x C-t  
 C-M-t

## Spelling Check

check spelling of current word  
 check spelling of all words in region  
 check spelling of entire buffer

M-\$  
 M-x ispell-region  
 M-x ispell-buffer

## Tags

find a tag (a definition)  
 find next occurrence of tag  
 specify a new tags file

M-.  
 C-u M-.  
 M-x visit-tags-table

rege~~xp~~ search on all files in tags table  
 run query-replace on all the files  
 continue last tags search or query-replace

M-x tags-search  
 M-x tags-query-replace  
 M-.,

## Shells

execute a shell command  
 run a shell command on the region  
 filter region through a shell command  
 start a shell in window \*shell\*

M-!  
 M-|  
 C-u M-|  
 M-x shell

## Rectangles

copy rectangle to register  
 kill rectangle  
 yank rectangle  
 open rectangle, shifting text right  
 blank out rectangle  
 prefix each line with a string  
 select rectangle with mouse

C-x r r  
 C-x r k  
 C-x r y  
 C-x r o  
 M-x clear-rectangle  
 M-x string-rectangle  
 M-button1

## Abbrevs

add global abbrev  
 add mode-local abbrev  
 add global expansion for this abbrev  
 add mode-local expansion for this abbrev  
 explicitly expand abbrev  
 expand previous word dynamically

C-x a g  
 C-x a l  
 C-x a i g  
 C-x a i l  
 C-x a e  
 M-/-

## Regular Expressions

any single character except a newline  
 zero or more repeats  
 one or more repeats  
 zero or one repeat  
 any character in the set  
 any character not in the set  
 beginning of line  
 end of line  
 quote a special character *c*  
 alternative ("or")  
 grouping  
*n*th group  
 beginning of buffer  
 end of buffer  
 word break  
 not beginning or end of word  
 beginning of word  
 end of word  
 any word-syntax character  
 any non-word-syntax character  
 character with syntax *c*  
 character with syntax not *c*

. (dot)  
 \*  
 +  
 ?  
 [...] ]  
 [...] ]  
 ^  
 \$  
 \|c  
 \|  
 \| ( ... \| )  
 \|n  
 \|'  
 \|'  
 \|  
 \|B  
 \|<  
 \|>  
 \|W  
 \|w  
 \|s  
 \|S  
 \|sc  
 \|Sc

## Registers

save region in register  
 insert register contents into buffer  
 save value of point in register  
 jump to point saved in register

C-x r s  
 C-x r i  
 C-x r i  
 C-x r j

## Info

enter the Info documentation reader  
 Moving within a node:  
 scroll forward  
 scroll reverse  
 beginning of node  
 Moving between nodes:  
 next node  
 previous node  
 move up  
 select menu item by name  
 select *n*th menu item by number (1-5)  
 follow cross reference (return with 1)  
 return to last node you saw  
 return to directory node  
 go to any node by name

Other:

run Info **tutorial**  
 list Info commands  
**quit** Info  
 search nodes for rege~~xp~~

M-/-

## Keyboard Macros

start defining a keyboard macro  
 end keyboard macro definition  
 execute last-defined keyboard macro  
 edit keyboard macro  
 append to last keyboard macro  
 name last keyboard macro  
 insert Lisp definition in buffer

C-x ( )  
 C-x )  
 C-x e  
 C-x C-k  
 C-u C-x ( )  
 M-x name-last-kbd-macro  
 M-x insert-kbd-macro

## Commands Dealing with Emacs Lisp

eval *sexp* before point  
 eval current defun  
 eval region  
 eval entire buffer  
 read and eval minibuffer  
 re-execute last minibuffer command  
 read and eval Emacs Lisp file  
 load from standard system directory

C-x C-e  
 C-M-x  
 M-x eval-region  
 M-x eval-current-buffer  
 M-ESC  
 C-x ESC ESC  
 M-x load-file  
 M-x load-library

## Simple Customization

Here are some examples of binding global keys in Emacs Lisp.

`(global-set-key [(control c)] 'goto-line)`  
`(global-set-key [(control x)] (control k)) 'kill-region)`  
`(global-set-key [meta #]) 'query-replace-regexp)`

An example of setting a variable in Emacs Lisp:  
`(setq backup-by-copying-when-linked t)`

## 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.  
 Negative counts from bottom."  
 (interactive "p")  
 (recenter (if (null line)  
 0  
 (prefix-numeric-value line))))`

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

Copyright © 1998 Free Software Foundation, Inc.  
 designed by Stephen Gildea, April 1998 v2.0 XEmacs  
 for GNU Emacs version 19 on Unix systems

Updated for XEmacs in February 1995 by Ben Wing

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.