

GNU Emacs Reference Card

by Bokyun Na (bkna@kpu.ac.kr)

Open/Quit: emacs file_name / C-x, C-c

Files

read a file into Emacs	C-x C-f
save a file back to disk	C-x s
insert another file into this buffer	C-x i
replace this with the file you really want	C-x C-v
write buffer to a specified file	C-x C-w
run dired, the directory editor	C-x d
copy	C compress/uncompress Z
rename	R change mode M
read file/view	f/v bitcompile B
delete	d then x help h
create directory	+

ftp/ssh

C-x C-f: /protocol:usr@host#port:dir/file

Motion

Cursor motion:

entity to move over	backward	forward
character	C-b	C-f
word	M-b	M-f
line	C-p	C-n
first of the line	C-a	C-e
sentence	M-a	M-e
paragraph	M-[M-]
page	C-x [C-x]
sexp	C-M-b	C-M-f
function(dfuction)	C-M-a	C-M-e

Screen motion:

scroll to next screen/to previous	C-v/M-v
scroll to center of screen	C-1
scroll left	C-x <
scroll right	C-x >

Editing

entity to kill	backward	forward
character(delete, not kill)	Del	C-d
delete word	M-Del	M-d
delete line(to end of)	M-o C-k	C-k
delete sentence	C-x Del	M-k
delete sexp	M-- C-M-k	C-M-k
kill region	C-w	
kill to next occurrence of char	M-x char	
delete black lines around point	C-x C-o	
delete all white space around point	M- \	
count lines in page/region	C-x/M-=	
go to a specific line	C-c l	
put exactly one space at point	M-SPC	
insert newline after point	C-o	
move rest of line vertically down	C-M-o	
yank back last thing killed(paste)	C-y	
replace last yank with previous kill	M-y	
copy	M-w, move corsor, C-y	
cut & paste	C-w, move cursor, C-y	

redo/undo

C-x ESC/C-x u

Formatting

indent current line(mode-dependent)	TAB
indent region(mode-dependent)	C-M- \
indent sexp(mode-dependent)	C-M-q
indent region rigidly arg columns	C-x TAB
fill paragraph	M-q
set fill column	C-x f
set prefix each line starts with	C-x

Spell Check

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

Search & Replace

search forward / backward	C-s / C-r
word search forward / backward	C-s M C-w/C-r M C-w
regular expression search	C-M-s
exit incremental search	ESC
delete incorrect searching characters	Del
abort current search	C-g
line search / page search	M-x what-line/M-x what-page

Interactively replace a text string (search&replace) M-%
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	n or Del
replace all remaining matches	!
back up to the previous match	^
exit query-replace	ESC
enter recursive edlt(C-M-c to exit)	C-r
exit recursive edit&exit query	C-]

using regular expressions M-x query-replace-regexp
move ^M from DOS file in search&replace, C-q C-m

Case Change

uppercase word/lower	M-u/M-l
capitalize word	M-c
uppercase region/lower	C-x C-u/C-x C-l
capitalize region	M-x capitalize-region

C&C++ : M-x C-mode

compile	M-x compile
comment column	M-;
Region comments	C-SPC M-;
insert new line	M-j
move to beginning of function body	M-c-a
move to end of function	M-c-e
cursor at beginning of function, mark at end	M-c-h
go to error line during compilation	C-x

Using a Mouse

paste(yank)	middle(S-right)
cut	right(S-middle)
cut and wipe	C-middle
split window vertically	C-right
delete other windows	C-S right
popup menu	C-SHIFT-left

Marking

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

Tags

Find tag	M-.
Find next occurrence of tag	C-u M-.
specify a new tags file	M-x visit-tags-table
regex search on all files in tags table	M-x tags-search
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*	C-z(M-x shell)
return to emacs	fg

Transposing

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

Multiple Windows

delete all other windows	C-x 1
delete this window	C-x 0
split window in 2 vertically	C-x 2
split window in 2 horizontally	C-x 3
scroll other window	C-M-v
switch cursor to another window	C-x o
shrink window shorter	M-x shrink-window
grow window taller	C-x ^
shrink window narrower	C-x {
grow window wider	C-x }
select a buffer in other window	C-x 4 b
find file in other window	C-x 4 f
Compose mail in other window	C-x 4 m
run Dired in other window	C-x 4 d
find tag in other window	C-x 4

Buffers

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

Print

print buffer with headers & page#	M-x print-buffer
print file directly without	M-x lpr-buffer
print region with	M-x print-region
print region without	M-x lpr-region

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	?
abort command	C-g

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

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

Error Recovery

abort partially typed of executing cammand	C-g
recover a file lost by a system crash	M-x recover-file
undo an unwanted change	C-? (C-x u or C-)
restore a buffer to its original contents	M-x revert-buffer
redraw garbaged screen	C-l
non-redraw garbaged screen search	C-s ESC"~"

Getting Help

Type C-h and follow the directions. If you are a first-time user, type **C-h t** for a tutorial.

get rid of Help window	C-x-1
scroll Help window	ESC C-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

Picture: M-x edit-picture

upward	C-c ^
downward	C-c .
right	C-c >
left	C-c <
northeast	C-c '
northwest	C-c `
southeast	C-c \
southwest	C-c /