## GNU Emacs Reference Card

**Starting Emacs**
To enter GNU Emacs 19, 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-x C-z
exit Emacs permanently C-x C-c
save all files C-x C-s
insert contents of another file into this buffer C-x i
replace this file with the file you really want C-x C-w
write buffer to a specified file C-x C-w

**Getting Help**
The help system is simple. Type C-h to follow the directions.
If you are a first-time user, type C-h t for a tutorial.

remote Help window C-x 1
scroll Help window C-x C-w
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-u
restore a buffer to its original contents M-x revert-buffer
redraw garbled screen C-l

**Incremental Search**
search forward C-s
search backward C-r
regular expression search C-x n
reverse regular expression search C-x n
select previous search string M-p
select next search string M-n
exit incremental search RET
undo effect of last character DEL
short cut 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.

<table>
<thead>
<tr>
<th>Motion</th>
<th>Multiple Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>entity to move over</td>
<td>delete all other windows</td>
</tr>
<tr>
<td>character</td>
<td>C-x 1</td>
</tr>
<tr>
<td>word</td>
<td>delete this window C-x 0</td>
</tr>
<tr>
<td>line</td>
<td>split window in two vertically C-x 2</td>
</tr>
<tr>
<td>go to line beginning (or end)</td>
<td>split window in two horizontally C-x 3</td>
</tr>
<tr>
<td>sentence</td>
<td>scroll other window C-x w</td>
</tr>
<tr>
<td>paragraph</td>
<td>switch cursor to another window C-x a</td>
</tr>
<tr>
<td>page</td>
<td>shrink window shorter M-x shrink-window</td>
</tr>
<tr>
<td>sexp</td>
<td>grow window taller C-x *</td>
</tr>
<tr>
<td>function</td>
<td>shrink window narrower C-x (</td>
</tr>
<tr>
<td>go to buffer beginning (or end)</td>
<td>grow window wider C-x )</td>
</tr>
<tr>
<td>scroll to next screen</td>
<td>select buffer in other window C-x 4 b</td>
</tr>
<tr>
<td>scroll to previous screen</td>
<td>display buffer in other window C-x o</td>
</tr>
<tr>
<td>scroll left</td>
<td>find file in other window C-x 4 f</td>
</tr>
<tr>
<td>scroll right</td>
<td>find file read-only in other window C-x 4 r</td>
</tr>
<tr>
<td>scroll current line to center of screen</td>
<td>run Direx in other window C-x 4 d</td>
</tr>
<tr>
<td>Killing and Deleting</td>
<td>find tag in other window C-x 4 .</td>
</tr>
<tr>
<td>entity to kill</td>
<td>Formatting</td>
</tr>
<tr>
<td>character (delete, not kill)</td>
<td>indent current line (mode-dependent) TAB</td>
</tr>
<tr>
<td>word</td>
<td>indent region (mode-dependent) C-x</td>
</tr>
<tr>
<td>line (to end of)</td>
<td>indent sexp (mode-dependent) C-x q</td>
</tr>
<tr>
<td>sentence</td>
<td>insert TAB C-x t</td>
</tr>
<tr>
<td>sexp</td>
<td>insert newline after point C-x</td>
</tr>
<tr>
<td>kill region</td>
<td>move rest of line vertically down C-x M-1</td>
</tr>
<tr>
<td>copy region to kill ring</td>
<td>delete blank lines around point C-x g</td>
</tr>
<tr>
<td>kill through next occurrence of char</td>
<td>join line with previous (with arg, next) M-1</td>
</tr>
<tr>
<td>yank back last thing killed</td>
<td>delete all white space around point M-1</td>
</tr>
<tr>
<td>replace last yank with previous kill</td>
<td>put exactly one space at point M-x SPC</td>
</tr>
</tbody>
</table>

**Case Change**
upper case word C-x U
lower case word C-x l
capitalize word C-x H
toupper region C-x C-q
tolower region C-x C-a
capitalize region M-x capitalize 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 ?
fetch previous minibuffer input M-p
fetch next minibuffer input M-n
regexp search backward through history M-r
regexp search forward through history M-s
abort command C-g
Type C-x C-s C-s 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
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-x e

Spelling Check

display spelling of current word M-s
display spelling of all words in region M-x ispell-region
display spelling of entire buffer M-x ispell-buffer

Tags

find a tag (a definition) C-t
find next occurrence of tag M-s M-t
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-s

Shells

execute a shell command N-s
run a shell command on the region N-s
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 to C-x r o
blank out rectangle M-x clear-rectangle
prefix each line with a string M-x string-rectangle

Abbrevs

add global abbrev C-x 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-/ a

Regular Expressions

any single character except a newline (dot)
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 \{ \}
g with group \]<
end of buffer 
word break not beginning or end of word \B
beginning of word \b
end of word \e
any word syntax character \w
any non-word syntax character \W
character with syntax c \c
character with syntax not c \C

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 p
jump to point saved in register C-x r j

Info

enter the Info documentation reader C-x i
Moving within a node:
scroll forward SPC
scroll reverse DEL
beginning of node .
Moving between nodes:
next node n
previous node p
move up u
select menu item by name m
select n-th menu item by number (1 5) n
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
list info commands ?
quit info q
search nodes for regexp s

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-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 e
current eval C-x e
region M-x eval-region
eval entire buffer C-x eval-buffer
read eval and eval macro M-x eval
execute last minibuffer command C-x ESC ESC
read and eval Emacs Lisp file M-x load-file
load from standard system directory M-x load-library

Simple Customization

Here are some examples of binding global keys in Emacs Lisp.
Note that you cannot say "M-\#": you must say "\#\#"

(global-set-key "C-c g" 'goto-line)
(global-set-key "C-c \C-k" 'kill-region)
(global-set-key "\C-k" 'query-replace-regexp)

An example of setting a variable in Emacs Lisp:
(setq backup-by-copying nil)

Writing Commands

(setq command-name (args)
"documentation"
(interactive "template")
body)
An example:
(setq this-line=1 to-top-of-window (line)
"With AGL, put point on line AGL."
(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 © 1993 Free Software Foundation, Inc.,
designed by Stephen Gilmore, May 1993 v2.0
for GNU Emacs version 19 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 copies.
For copies of the GNU Emacs manual, write to the Free Software
Foundation, Inc., 512 Massachusetts Avenue, Cambridge, MA 02139.