

GNU Emacs Reference Card

(for version 20)

Starting Emacs

To enter GNU Emacs 20, just type its name: `emacs`

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

Leaving Emacs

Suspend Emacs (or icontty it under X)

exit Emacs permanently

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

version control checkout/checkout

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

scroll help window

apropos: show commands matching a string

show the function a key runs

describe a function

get mode-specific information

Error Recovery

abort partially typed or executing command

recover a file lost by a system crash

undo an unwanted change

restore a buffer to its original contents

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

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

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

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

Marking

set mark here

exchange point and mark

set mark *any words* away

mark *paragraph*

mark *page*

mark *sexp*

mark *function*

mark entire buffer

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)

backward forward

`C-f`

`M-b`

`C-p`

`C-a`

`M-a`

`M-}`

`C-x [`

`C-M-f`

`M-<`

`C-v`

`M-v`

`C-x <`

`C-x >`

`C-u C-l`

backward forward

`DEL`

`M-DEL`

`M-0 C-k`

`C-x DEL`

`M-- C-M-k`

`C-w`

`M-w`

`M-z char`

`C-y`

`M-y`

`M-%`

`M-x query-replace-regex`

`SPC`

`,`

`DEL`

`i`

`~`

`RET`

`C-r`

Multiple Windows

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

delete all other windows

split window, above and below

delete this window

split window, side by side

scroll other window

switch cursor to another window

select buffer in other window

display buffer in other window

find file read-only in other window

run Dired in other window

find tag in other window

grow window taller

shrink window narrower

grow window wider

grow window wider

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

`C-x }`

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 later minibuffer input or default

regex search backward through history

abort command

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

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 later minibuffer input or default

regex search backward through history

abort command

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

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

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

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

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

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

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

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

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

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

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 *c* *c*
alternative ("or") \|
grouping *(...)*
same text as *n*th group *n*
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>	\sc	\Sc

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 c
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 *n*th menu item by number (1-9) 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
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 20, June 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