

# 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 iconify 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 checkin/checkout

C-x C-f  
C-x C-s  
C-x s

C-x i

C-x C-v

C-x C-w

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

scroll help window

apropos: show commands matching a string

show the function a key runs

describe a function

get mode-specific information

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

undo an unwanted change

restore a buffer to its original contents

redraw garbagged screen

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

## Incremental Search

search forward

regular 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

C-b

M-b

C-p

C-a

M-a

M-|

C-x [

C-M-b

C-M-a

M-<

C-f

M-f

C-n

C-e

M-e

M-]

C-x ]

C-M-f

C-M-e

M->

C-v

M-v

C-x <

C-x >

C-u C-1

## 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  
DEL  
M-DEL  
M-0 C-k  
C-x DEL  
M-- C-M-k  
C-M-k

forward

C-d

M-d

C-k

C-x DEL

M-k

C-M-k

C-w

M-w

M-z char

C-y

M-y

## 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-e to exit)

M-%  
M-x query-replace-regex

SPC

,

DEL

!

-

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 in other window

find file read-only in other window

run Dirred in other window

find tag in other window

grow window taller

shrink window narrower

grow window wider

C-x 1

C-x 2

C-x 0

C-x 3

C-M-v

C-x o

C-x 4 b

C-x 4 C-o

C-x 4 f

C-x 4 r

C-x 4 d

C-x 4 .

C-x `

C-x {

C-x }

## 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

set face

TAB

C-M-\

C-M-q

C-x TAB

C-o

C-M-o

C-x C-o

M-^

M-\

M-SPC

M-q

C-x f

C-x .

M-g

## Case Change

uppercase word

lowercase word

capitalize word

uppercase region

lowercase region

M-u

M-l

M-c

C-x C-u

C-x C-l

## 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 later minibuffer input or default

regex search backward through history

regex search forward through history

abort command

TAB

SPC

RET

?

M-p

M-n

M-r

M-s

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

## Regular Expressions

any single character except a newline  
 zero or more repeats  
 one or more repeats  
 zero or one repeat  
 quote regular expression special character *c*  
 alternative ("or")  
 grouping  
 same text as *n*th group  
 at word break  
 not at word break  
 entity  
 line  
 word  
 buffer  
 class of characters  
 explicit set  
 word-syntax character  
 character with syntax *c*

.	(dot)
*	zero or more repeats
+	one or more repeats
?	zero or one repeat
\c	quote regular expression special character <i>c</i>
\	alternative ("or")
\( ... \)	grouping
\n	same text as <i>n</i> th group
\b	at word break
\B	not at word break
entity	entity
line	line
word	word
buffer	buffer
class of characters	class of characters
explicit set	explicit set
word-syntax character	word-syntax character
character with syntax <i>c</i>	character with syntax <i>c</i>

## International Character Sets

specify principal language  
 show all input methods  
 enable or disable input method  
 set coding system for next command  
 show all coding systems  
 choose preferred coding system

M-x set-language-environment	M-x set-language-environment
M-x list-input-methods	M-x list-input-methods
C-\	enable or disable input method
C-x RET c	set coding system for next command
M-x list-coding-systems	M-x list-coding-systems
M-x prefer-coding-system	M-x prefer-coding-system

## Info

enter the Info documentation reader  
 find specified function or variable in Info  
 Moving within a node:  
 scroll forward  
 scroll reverse  
 beginning of node  
 Moving between nodes:

C-h i	enter the Info documentation reader
C-h C-i	find specified function or variable in Info
SPC	scroll forward
DEL	scroll reverse
.	beginning of node
(dot)	beginning of node

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

## Keyboard Macros

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

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

## Commands Dealing with Emacs Lisp

eval sexp before point  
 eval current defun  
 eval region  
 read and eval minibuffer  
 load from standard system directory

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

## Simple Customization

customize variables and faces  
 Making global key bindings in Emacs Lisp (examples):  
 (global-set-key "\C-cg" 'goto-line)  
 (global-set-key "\M-#" 'query-replace-regex)

## 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

## Buffers

select another buffer  
 list all buffers  
 kill a buffer

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

## Transposing

transpose characters  
 transpose words  
 transpose lines  
 transpose sexps

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

## Spelling Check

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

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

## Tags

find a tag (a definition)  
 find next occurrence of tag  
 specify a new tags file  
 regepx search on all files in tags table  
 run query-replace on all the files  
 continue last tags search or query-replace

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

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

## Rectangles

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

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

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