

# GNU Emacs Reference Card

## 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) **C-z**  
exit Emacs permanently **C-x C-c**

## Files

**read** a file into Emacs **C-x C-f**  
**save** a file back to disk **C-x C-s**  
**save all files** **C-x s**  
**insert** contents of another file into this buffer **C-x i**  
replace this file with the file you really want **C-x C-v**  
write buffer to a specified file **C-x C-w**  
version control checkin/checkout **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 **C-x l**  
scroll help window **C-M-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**

## 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-\_-**  
restore a buffer to its original contents **M-x revert-buffer**  
redraw garbagged screen **C-l**

## Incremental Search

search forward **C-s**  
search backward **C-r**  
regular expression search **C-M-s**  
reverse regular expression search **C-M-r**  
select previous search string **M-p**  
select next later search string **M-n**  
exit incremental search **RET**  
undo effect of last character **DEL**  
abort current 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.

## Motion

**entity to move over**  
character **C-b**  
word **M-b**  
line **C-P**  
go to line beginning (or end) **C-a**  
sentence **M-a**  
paragraph **M-|**  
page **C-x l**  
sexp **C-M-b**  
function **C-M-a**  
go to buffer beginning (or end) **M-<**  
scroll to next screen **C-v**  
scroll to previous screen **M-v**  
scroll left **C-x <**  
scroll right **C-x >**  
scroll current line to center of screen **C-u C-l**

## Killing and Deleting

**entity to kill**  
character (delete, not kill) **backward** **forward**  
word **DEL** **C-d**  
line (to end of) **M-DEL** **M-d**  
sentence **M-0 C-k** **C-k**  
sexp **C-x DEL** **M-k**  
**kill region** **C-w**  
copy region to kill ring **M-w**  
kill through next occurrence of *char* **M-z char**  
yank back last thing killed **C-y**  
yank back last yank with previous kill **M-y**

## Marking

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

## Query Replace

interactively replace a text string **M-%**  
using regular expressions **M-x query-replace-regex**  
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 **DEL**  
**replace** all remaining matches **!**  
**back up** to the previous match **-**  
**exit** query-replace **RET**  
enter recursive edit (**C-M-e** to exit) **C-r**

## Multiple Windows

When two commands are shown, second is for "other frame."  
delete all other windows **C-x l**  
split window, above and below **C-x 2** **C-x 5 2**  
delete this window **C-x 0** **C-x 5 0**  
split window, side by side **C-x 3**  
scroll other window **C-M-v**  
switch cursor to another window **C-x o** **C-x 5 o**  
select buffer in other window **C-x 4 b** **C-x 5 b**  
display buffer in other window **C-x 4 C-o** **C-x 5 C-o**  
find file in other window **C-x 4 f** **C-x 5 f**  
find file read-only in other window **C-x 4 r** **C-x 5 r**  
run Dired in other window **C-x 4 d** **C-x 5 d**  
find tag in other window **C-x 4 .** **C-x 5 .**  
grow window taller **C-x ^**  
shrink window narrower **C-x {**  
grow window wider **C-x }**

## 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**  
insert newline after point **C-o**  
move rest of line vertically down **C-M-o**  
delete blank lines around point **C-x C-o**  
join line with previous (with arg, next) **M-^**  
delete all white space around point **M-\**  
put exactly one space at point **M-SPC**  
fill paragraph **M-q**  
set fill column **C-x f**  
set prefix each line starts with **C-x .**  
set face **M-g**

## Case Change

uppercase word **M-u**  
lowercase word **M-l**  
capitalize word **M-c**  
uppercase region **C-x C-u**  
lowercase region **C-x C-l**

## 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 later minibuffer input or default **M-n**  
regeXP search backward through history **M-r**  
regeXP search forward through history **M-s**  
abort command **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	.	(dot)
zero or more repeats	*	
one or more repeats	+	
zero or one repeat	?	
quote regular expression special character <i>c</i>	\c	
alternative ("or")	\	
grouping	\( ... \)	
same text as <i>n</i> th group	\n	
at word break	\b	
not at word break	\B	
<b>entity</b>		
line	~	match start
word	<	>
buffer	\'	\'
<b>class of characters</b>		
explicit set	[ ... ]	[ ... ]
word-syntax character	\w	\W
character with syntax <i>c</i>	\sc	\Sc
<b>match these</b>		
<b>match end</b>		
<b>match others</b>		

## 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 regeexp         **M-s**

## 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 sexps

**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  
 regeexp search on all files in tags table  
 run query-replace on all the files  
 continue last tags search or query-replace

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

**C-x r r**  
**C-x r k**  
**C-x r y**  
**C-x r o**  
**C-x r c**  
**C-x r t**

## 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-/**

## 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 SPC**  
**C-x r j**

## 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 (**  
**C-x )**  
**C-x e**  
**C-u C-x (**  
**M-x name-last-kbd-macro**  
**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-regeexp)

## 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