GNU Emacs Reference Card

(for version 24)

Starting Emacs

To enter GNU Emacs 24, just type its name: emacs

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	
insert contents of another file into this buffer C	-xi	
replace this file with the file you really want $\mathtt{C-x}$	C-v	
write buffer to a specified file	C-x	C-M
toggle read-only status of buffer	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	C-x 1 C-M-v
apropos: show commands matching a string	C-h a
describe the function a key runs	C-h k
describe a function	C-h f
get mode-specific information	C-h m

Error Recovery

abort partially typed or executing com-	mand C-g
recover files lost by a system crash	M-x recover-session
undo an unwanted change	C-x u, C or C-/
restore a buffer to its original contents	M-x revert-buffer
redraw garbaged screen	C-1

Incremental Search

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

c 2015 Free Software Foundation, Inc. Permissions on back

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)	backwa C-b M-b C-p C-a M-a M-{ C-x [C-M- b C-M- a M-<		forward C-f M-f C-n C-e M-e M-} C-x] C-M-f C-M-e M->
scroll to next screen scroll to previous screen scroll left scroll right scroll current line to center, top, bott goto line goto char back to indentation	om	C-v M-v C-x C-x C-1 M-g M-g M-m	> g

Killing and Deleting

entity to kill	backward	forward
character (delete, not kill)	DEL	C-d
word	M-DEL	M-d
line (to end of)	M-0 C-k	C-k
sentence	C-x DEL	M-k
sexp	M C-M-k	C-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 replace	С-у	
last yank with previous kill	М-у	

Marking

set mark here	C-@ or C-SP
exchange point and mark	C-x C-x
set mark arg words away	M-@
mark paragraph	M-h
mark page	С-х С-р
mark sexp	C-M-@
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 m	ıode	are
replace this one, go on to next		SPC or y
replace this one, don't move		
skip to next without replacing		DEL or n
replace all remaining matches		
back up to the previous match		
exit query-replace		RET
enter recursive edit (C-M-c to exit)		C-r

Multiple Windows

when two commands are shown, the s	econ.	ıu .	is a s	шша	I	
com-mand for a frame instead of a win	.dow					
delete all other windows	C-x			C-x		
split window, above and below	C-x			C-x		2
delete this window	C-x	0		C-x		0
split window, side by side			C-x			
scroll other window			C-M-			
switch cursor to another window	C-x			C-x		0
select buffer in other window	C-x			C-x		b
display buffer in other window find	C-x		C-o	C-x		С
file in other window	G-x			C-x		f
find file read-only in other window	C-x			C-x		r
run Dired in other window	C-x			C-x		d
find tag in other window	C-x			C-x		
grow window taller			C-x			
shrink window narrower			C-x			
grow window wider			C-~			

Formatting

indent current line (mode-dependent)	TAB
indent region (mode-dependent)	C-M-\
indent sexp (mode-dependent) indent	C-M-q
region rigidly arg columns indent for	C-x TAB
comment	M-;
insert newline after point	C-o
move rest of line vertically down delete	C-M-o
blank lines around point	C-x C-o
join line with previous (with arg, next)	M-^
delete all white space around point put	M-\
exactly one space at point	M-SPC
fill paragraph	M-q
set fill column to arg	C-x f
set prefix each line starts with	C-x .
set face	M-o

Case Change

uppercase word	M-u
lowercase word	M-1
capitalize word	M-c
uppercase region	C-x C-u
lowercase region	C-x C-1

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	М-р
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 he minibuffer. Type F10 to activate menu bar items on text erminals.

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 check		M-\$
spelling of all words in region check	M-x	ispell-region
spelling of entire buffer toggle on-	M-x	ispell-buffer
the-fly spell checking	M-x	flyspell-mode

Tags

find a tag (a definition) find	M
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-re	eplace M-,

Shells

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

Rectangles

copy rectangle to register	C-x r r
kill rectangle	C-x r k
yank rectangle	С-х г у
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 add	C-x a i g
mode-local expansion for this abbrev	C-x a i l
explicitly expand abbrev	C-x a e
expand previous word dynamically	M-/

Miscellaneous

numeric argument	C-u	
negative argument	M	
quoted insert	C-q	char

Regular Expressions

any single character except	a newline	\cdot (dot)
zero or more repeats		
one or more repeats		
zero or one repeat		
quote special characters		
quote regular expression spe	ecial character c	
alternative ("or")		\
grouping		\(\)
shy grouping		\(:? \)
explicit numbered grouping		\(:NUM \)
same text as nth group		$\setminus n$
at word break		\b
not at word break		\ B
entity	match start	match end
		\$
word	\<	\>
symbol		
buffer		
class of characters	match these	match others
explicit set word-		
syntax character	\w	\W
character with syntax c	\sc	\S <i>c</i>
character with category c	\cc	\Cc

International Character Sets

specify principal language	C-x RET 1
show all input methods	M-x list-input-methods
enable or disable input method	C-\
set coding system for next comma	and C-x RET c
show all coding systems choose	M-x list-coding-systems
preferred coding system	M-x prefer-coding-system

Info

enter the Info documentation reader find	C-h i
specified function or variable in Info	C-h S
Moving within a node:	
scroll forward	SPC
scroll reverse	DEL
beginning of node	
Moving between nodes:	
next node	
previous node	
move up	
select menu item by name	
select n th menu item by number $(1-9)$	
follow cross reference (return with 1)	
return to last node you saw	
return to directory node	
go to top node of Info file	
go to any node by name	g
Other:	
run Info tutorial	
look up a subject in the indices	
search nodes for regexp	
quit Info	q

Registers

	ave region in register	C-x		
	sert register contents into buffer	C-x		
	ave value of point in register	C-x		SPC
įι	imp to point saved in register	C-x	r	

Keyboard Macros

start defining a keyboard macro	C-x (
end keyboard macro definition	C-x)
execute last-defined keyboard mac	ro 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		С-х С-е
eval current defun		C-M-x
eval region	M-x	eval-region
read and eval minibuffer		M-:
load a Lisp library from load-path	M-x	load-library

Simple Customization

```
customize variables and faces M-x customize
Making global key bindings in Emacs Lisp (example):

(global-set-key (kbd "C-c g") 'search-forward)
(global-set-key (kbd "M-#") 'query-replace-regexp)
```

Writing Commands

(defun command-name (args)

The interactive spec says how to read arguments interactively. Type C-h f interactive RET for more detail

Copyright c○ 2015 Free Software Foundation, Inc For GNU Emacs version 24 Designed by Stephen Gildea

Permission is granted to make and distribute modified or unmodified copies of this card provided the copyright notice and this permission notice are preserved on all copies.

For copies of the GNU Emacs manual, see:

http://www.gnu.org/software/emacs/#Manuals