GNU Emacs Reference Card Reloaded

(for version 27)

Starting Emacs
To enter GNU Emacs 27, 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
open file at (relative or abs.) point	M-x ffap
open file at (relative or abs.) point in another window	M-x ffaow
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
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	C-x 1
scroll help window	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 command	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-l

Search
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 matched.

Occur search

List occurrences	M-s o
(creates new window)	
List occurrences of word at point	M-s .

Miscellaneous
Toggle case sensitivity of search	M-x toggle-case-fold-search

Dired (C-x d)
Go to next/previous entry	n / p
Create directory	+
Cd up	^
Open file	RET / f
Open file in other window	o
Open file in other windows, no select	C-o
Mark entry	m
Mark entry defined by regexp	%m
Mark all files by grep-ing content	%g
Flag entry for deletion	d
Flag entry defined by regexp	%d
Promp for copy location	C
Apply *mv* to entry	R
Create hard link	H
Create symbolic link	S
Change entry permission	M
Unflag files	u
Confirm and delete flagged entries	x
Execute a command on selected entries	!
Idem, asynchronously	&
Compare entry to	=
Query regexp replace in selected entries	Q
Regexp occur in selected entries	

Motion
entity to move over	backward	forward
character	C-b	C-f
word	M-b	M-f
line	C-p	C-n
go to line beginning (or end)	C-a	C-e
sentence	M-a	M-e
paragraph	M-{	M-}
page	C-x [	C-x]
sexp	C-M-b	C-M-f
function	C-M-a	C-M-e
go to buffer beginning (or end)	M-<	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, top, bottom		C-l
goto line		M-g g
goto char		M-g c
back to indentation		M-m
goto prior mark (see Marking)		C-u C-SPC

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
If no region selected, copy line.	
kill through next occurrence of *char*	M-z *char*
yank back last thing killed	C-y
replace last yank with previous kill	M-y
*Italicised: disable **smartparens** first.*	

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

Query Replace
(current buffer)
interactively replace a text string	M-%
using regular expressions	M-x query-replace-regexp / C-M-%

Valid responses in query-replace mode 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 second is a similar command for a frame instead of a window.

delete all other windows	C-x 1	C-x 5 1
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

S-<LEFT/RIGHT/UP/DOWN>

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

C-S-<LEFT/RIGHT/UP/DOWN>

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 / smaller	(C-u N) C-x ^ / C-u -N 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
indent for comment/uncomment	M-;

Comment region	C-c C-c
uncomment region	C- - C-c C-c
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
Cut cooment at point	M-j
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-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 menu bar items on text terminals.

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 toggle on-the-fly
spell checking	M-x flyspell-mode

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-,
Find references	M- ?
List all tags	M-x list-table

Shells

execute a shell command	M-!
execute a shell command asynchronously	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-/

Miscellaneous

numeric argument	C-u *num*
negative argument	M--
quoted insert	C-q char
Repeat latest command (N times)	(C-u N) C-x z then z
Man	M-x man/woman
Calendar	M-x calendar

Regular Expressions

any single character except a newline	. (dot)
zero or more repeats	*
one or more repeats	+
zero or one repeat	?
quote special characters	\
quote regular expression special character *c*	\c
alternative ("or")	|
grouping	\(... \)
shy grouping	\(:? ... \)
explicit numbered grouping	\(:NUM ... \)
same text as *n*th group	\n
at word break	\b
not at word break	\B

entity	match start	match end
line	^	$
word	\<	\>
symbol	_<	_>
buffer	\'	\'
class of characters	**match these**	**match others**
explicit set	[...]	[^ ...]
word-syntax character	\w	\W
character with syntax *c*	\s*c*	\S*c*
character with category *c*	\c*c*	\C*c*

International Character Sets

specify principal language	C-x RET l
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 S

Moving within a node:

scroll forward	SPC
scroll reverse	DEL
beginning of node	b

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 l)	f
return to last node you saw	l
return to directory node	d
go to top node of Info file	t
go to any node by name	g

Other:

run Info **tutorial**	h
look up a subject in the indices	i
search nodes for regexp	s
quit Info	q

Registers and bookmarks

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
save point in named bookmark	C-x r m bookmark
jump to point saved in register / bookmark	C-x r j / b
List all bookmarks	C-x r l
Save current bookmark values	M-x bookmark-save

Keyboard Macros

start defining a keyboard macro	C-x (| F3
end keyboard macro definition	C-x) | F4
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
Apply last keyboard macro to all	C-x C-k r
lines in the region.	apply-macro-to-region-lines

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 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*) "*documentation*"
    (interactive "*template*") *body*)

An example:

```
(defun this-line-to-top-of-window (line)
    "Reposition current line to top of window.
With prefix argument LINE, put point on LINE."
    (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 RET for more details.

C-type Navigation

Move forward over a balanced expression	C-M-f
Move backward over a balanced expression	C-M-b
Kill balanced expression (sexp)	C-M-k
Put mark after sexp	C-M-SPC
Move point to beginning of function	C-M-a
Move point to end of function	C-M-e
Select current or following funtion	C-M-h
Jump to definitions/headers/references	M- .
	M-x helm-gtags-dwim
Jump back to original location after M- .	M- ,
Find references/symbols/function calls/pattern	C-c g r / s / a /g
List visited tags	C-c g h
Inner helm navigation :	C-n / p
	C-j / RET

Speedbar

Open children of node	SPC
Open node in another window	RET
Go to parent directory	U
Move to next or previous node	n / p
Move to next or previous node at current level	M-n / M-p
Switch to buffer list using Speedbar presentation	b
Switch back to file tree	f

Senator and semantic (prefix : C-c ,)

Next/Previous tag	n / p
Tag up	u
Copy/cut tag	M-w / C-w
Go to reference tag	g
Inner navigation	
Nest/previous	n / p
Go to/show	RET / SPC
Locally unfold/fold tag	+ / -
Globally unfold/fold tags	C-c C-e / C-c C-r

Projectile (prefix : C-c p-)

Jump to any project file	f
List files in project	F
List files/buffers/projects	h
	M-x helm-projectile
Jump to any project directory	d
List project buffers	b
Jump to recent files	e
List files	g
Grep in project	s g
Multi-occur in project buffers	o
Open project buffer	(4) C-o
(in other window)	
Text replace in current project	r
Find reference	j
Prior/next buffer	
LEFT/RIGHT	
Switch projects	p
Switch to recent project	ESC
Compile at project root	c
Configure at project root	C
Buffer list by recentcy	I
Switch source and header	a
Select projectile command	M/m
Save all project buffers	S
Regenerate TAGS file	R
Projectile command list	C-h

Compilation *(M-x (re)compile)*

Display matched location, do not switch point	C-o
Move to next error message, do not visit error	M-n
Move to previous error, do not visit location	M-p
Move to next error message, visit error location	M-g n
Move to previous error message, visit error location	M-g p
Visit location of error at point	RET
Move point to previous error in other file	M-{
Move point to next error in other file	M-}
Quit compilation buffer	q
Compile again (compilation buffer)	g
Abort compilation	M-x kill compilation

Fold / Unfold (prefix : C-c @-)

M-x hs-minor-mode

Hide/show current block	C-h / s
Toggle show lower blocks	C-l
Hide/show all blocks but first lines	C-M-h / s

Narrowing (prefix : C-x n)

Hide text below point	n
Narrow to current def	d
Un-narrow (widen)	w

Company-mode

	M-n / p
	C-w
	F1
	M-digit
	TAB

Magit (launch : C-x g)

c [c	a	f]	Prepare for commit	amend	fixup
C-c [C-c	C-k]	Commit	Kill commit
P [-F] p	Push [--force]
F [p|u|e]	Pull [remote|upstream|other]
b l	Change branch
$	View git console
M-p/f	Previous/Next com. msg

Helm (prefix : C-c h)

Google suggests	g
Color picker	c
Imenu	i
Inner navigation :	C-n / p / j /g
next / previous / select / quit	
Occur search in current buffer	M-s o
(splits window vertically)	
Grep in buffer	C-c h r
Helm mark ring	C-h SPC

Custom tweaks

Split window horizontally	F2
Set mark and deactivate region	F7 (= C-SPC C-SPC)
Circle back mark ring	F8 (=C-u C-SPC)
Circle back mark ring 2 steps	F9 (=C-u C-SPC C-u C-SPC)
Man at point	F11
Get to latest mark [and back]	M-x get-back	C-u C-x C-x
Apply dos2unix to current buffer	M-x d2u
Flush out double blank lines	M-x d2e
Apply *dos2unix* to currentbuffer and flush out double blank lines	M-x ddd
Move end-of-line comment above line blankline	M-x move-comment-up-
Run bash on region	C-M- !
	M-x run-bash-on-region
Recompile	C-M-y
Next buffer	C-*
Previous buffer	C-$
Render HTML in current buffer	eww-rcb
Compile your current file via	C-c C-c
Compile and run your current file (no interactive input)	C-c C-r
Do not ask about saving buffers before compile AND save all buffers	(setq compilation-ask-about-save nil)

For more Emacs documentation, and the TEX source for this card,
see the Emacs distribution, or
https://www.gnu.org/software/emacs