

Long Emacs Reference Card

open

or click a graphical icon `emacs& RET`
 start in terminal, not in a windowing system

information about Emacs `emacs -nw RET`
 show version in echo area `C-h C-a`
 show time since started `M-x emacs-version`
`M-x emacs-uptime`

exit

save buffers and quit `C-x C-c`
 force save buffers and quit `C-u C-x C-c`
 recursive edit (see "recursive edit" below)

force quit `C-]`
`M-x kill-emacs`

cancel

a command `C-g`
 recursive edit (see "recursive edit" below)

same as previous `C-M-c`
`C-[C-c`

recursive edit

exit recursive edit `C-]`
 cancel recursive edit `C-M-c`
 same as previous `C-[C-c`
 cancel all `M-x top-level`

customize

Emacs initialization file `C-x C-f ~/.emacs RET`

run time of loading user init file `M-x emacs-init-time`
`M-x customize`

main menu variable `M-x customize-variable`
 search mode `M-x customize-apropos`
`M-x customize-mode`

define key binding (see "key" below) `M-x global-set-key`

define key binding for current buffer `M-x local-set-key`

reread settings in file local variables `M-x normal-mode`

show clock, system load and email flag in mode line `M-x display-time`

turn off `M-x display-time-mode`

show times around the world `M-x display-time-world`
 show system power `M-x display-battery-mode`

show size in mode line `M-x size-indication-mode`
 show column number in mode line `M-x column-number-mode`

show end of file in fringe `M-x toggle-indicate-empty-lines`

add a ruler to the current buffer's window `M-x ruler-mode`

toggle existence of drop-down menu

toggle existence of clickable tool bar `M-x menu-bar-mode`

toggle scroll bar `M-x tool-bar-mode`
 toggle scroll bar in current frame `M-x scroll-bar-mode`

toggle blinking of cursor `M-x toggle-scroll-bar`
 fix delete and backspace keys `M-x blink-cursor-mode`

`M-x normal-erase-is-backspace-mode`

menu

text interaction with drop-down menu `M-'`
 same as previous `F10`
 toggle existence of drop-down menu

toggle existence of clickable tool bar `M-x menu-bar-mode`

`M-x tool-bar-mode`

help

menu `C-h ?`
 menu `C-h C-h`
 scroll down menu `SPC`

scroll up menu `DEL`
 close menu `q`
 tutorial `C-h t`

Emacs info manual (see "info" below) `C-h r`
 Emacs FAQ `C-h F`

what is command for KEY `C-h c <KEY>`
 describe command for KEY `C-h k <KEY>`
 where is key binding for COMMAND `C-h w <COMMAND> RET`

show current modes `C-h m`
 show current key bindings `C-h b`
 list bindings starting with C-x (see "key" below)

list bindings starting with C-c (see "key" below) `C-x C-h`

list commands matching search `C-c C-h`
 list commands and functions matching search `C-h a`

describe FUNCTION `C-u C-h a`
 describe last command executed with M-x `C-h f <FUNCTION> RET`

describe and show values for VARIABLE `M-x M-p C-h f RET`

list user variable matching search `C-h v <VARIABLE> RET`

list all variables matching search `M-x apropos-variable`

list user variables with values matching search `C-u M-x apropos-variable`

list all variables with values matching search `M-x apropos-value`
 list all with values matching search `C-u M-x apropos-value`

forward to cross-reference link `C-u M-x apropos-documentation`
 backward cross-reference link `TAB`
 same as previous `S-TAB`
 same as previous `M-TAB`
 same as previous `C-M-i`
 follow cross-reference `C-[C-i`
 follow cross-reference `RET`
 go back `C-c C-c`
 quit `C-c C-b`
`q`

info help

open directory of manuals `C-h i`
 open Emacs manual `C-h r`
 find description of symbol in manual `C-h S`

close `q`
 table of contents (menu) `t`
 back to directory `d`

visit menu ENTRY in table of contents `m <ENTRY>`
 forward to cross-reference link `TAB`
 backward cross-reference link `S-TAB`

same as previous `M-TAB`
 same as previous `C-M-i`
 same as previous `C-[C-i`

follow link `RET`
 back to last visited page `l`
 forward to last visited page `r`

scroll down `C-v`
 scroll down `SPC`
 scroll up `M-v`

scroll up `DEL`
 scroll up `b`
 next node `n`

previous node `p`
 search current manual's index `i`
 go to index node `i RET`

search all indices `M-x info-apropos`
 search forward for regular expression PATTERN `s <PATTERN>`

case-insensitive search for regular expression PATTERN `S <PATTERN>`

forward to SEARCH (see "search" below) `C-s <SEARCH>`
 reverse to SEARCH `C-r <SEARCH>`
 make a duplicate buffer in other window (see "window" below) `M-n`

same as previous `M-x clone-buffer`
 kill current node name as copy `c`
 same as previous `w`

kill current node name as Lisp expression `M-0 c`
 same as previous `M-0 w`

minibuffer

previous input `M-p`
 recent input `M-n`
 complete name of buffer, file, symbol (see "completion" below)

`TAB`
 same as previous `C-i`
 search previous input backward with REGEXP

search for previous input forward with REGEXP M-r <REGEXP> RET
 search previous input backward again M-s <REGEXP> RET
 search for previous input again M-r RET
 show recently echoed messages M-s RET
 exit C-h e
 C-g

completion

complete name of buffer, file, function, variable, ...

TAB
 SPC
 completion, unless a file C-q SPC
 insert space without completion ?
 list completions M-v
 go to "completions" buffer <right>
 next completion <left>
 previous completion RET
 select completion ESC ESC ESC
 exit back to minibuffer C-[C-[C-[
 same as previous

mode

help with current (see "help" above) C-h m
 for writing M-x text-mode
 a simple default M-x fundamental-mode
 change back to what Emacs thought it was M-x normal-mode
 customize current (see "customize" above) M-x customize-mode

add to top of file

M-x add-file-local-variable-prop-line RET mode RET

file

open C-x C-f
 new C-x C-f
 open in parent directory C-x C-f .. <file> RET
 open from root directory C-x C-f / <file> RET
 open all C sources in current directory C-x C-f *.c RET

including parent directories

M-x make-directory RET RET

close current and open another C-x C-v
 save C-x C-s
 save even if buffer is unchanged C-x C-w RET
 same as previous (see M-~below)

C-u M-~C-x C-s
 prompt to save any buffer that has been modified

C-x s
 save any buffer that has been modified with no questions

C-u C-x s
 open previously saved or opened C-x C-f M-p RET
 save current to PATH C-x C-w <PATH> RET
 save to DIRECTORY using file or buffer name

C-x C-w <DIRECTORY> RET

mark buffer as unmodified M-~
 mark buffer as modified C-u M-~

toggle as read only C-x C-q
 list contents of archive C-x C-f archive.tar RET
 open compressed Gzip C-x C-f file.gz RET
 list contents of ZIP C-x C-f file.zip RET
 rename current M-x rename-file RET <FILE> RET <NEW>
 rename current to NAME M-x rename-file RET RET <NAME> RET
 rename current to NAME

M-x rename-file RET RET <NAME> RET
 delete NAME M-x delete-file RET <NAME> RET
 (see "read only" below) M-x view-file
 open in basic mode with no conversions

M-x find-file-literally
 toggle between overwrite mode and insert mode

<insert>
 same as previous M-x overwrite-mode
 edit as literal bytes M-x binary-overwrite-mode
 rename previously saved or opened C-x d M-p RET R
 insert other into current buffer C-x i
 save region (see "region" below) M-x write-region
 save buffer once to alternate

C-x h M-x write-region
 save region to other (see "region" below)

M-x append-to-file
 set buffer to alternate M-x set-visited-file-name
 set permissions of current M-x set-file-modes
 open name at point M-x find-file-at-point
 restore buffer from disk M-x revert-buffer
 same as previous C-x C-v RET
 same as previous C-x C-f M-n RET
 open backup C-x C-f M-n ~RET
 save and make it the next backup C-u C-x C-s
 save without making a backup C-0 C-x C-s
 backup previous then save C-u C-u C-x C-s
 backup previous, save, make it the next backup

C-u C-u C-u C-x C-s

recover auto-save data after a crash M-x recover-file
 recover using auto-save data M-x recover-session
 show size in mode line M-x size-indication-mode
 update with changes on disk M-x auto-revert-mode
 update with changes for any buffer

M-x global-auto-revert-mode
 update end of buffer with changes on disk

M-x auto-revert-tail-mode
 use color overlays for tracking edits

M-x highlight-changes-mode
 make link M-x add-name-to-file
 make symbolic link M-x make-symbolic-link
 flush changes in file system to disk M-x unix-sync

buffer

beginning M-<
 end M->
 go to half way M-5 M-<
 go to 10 percent M-1 M-<

go to 90 percent M-9 M-<
 go back (see "mark" below) C-u C-SPC
 mark C-x h
 same as previous M-> M-<
 kill BUFFER C-x k <BUFFER> RET
 kill current C-x k RET
 ask to kill each and all M-x kill-some-buffers
 automatically kill old buffers

M-x clean-buffer-list
 schedule daily cleanup

M-x customize-group RET midnight RET
 switch to last buffer C-x b RET
 switch to BUFFER or make new BUFFER

C-x b <BUFFER> RET
 switch to a buffer in other window (see "window" below)

C-x 4 b
 show a buffer in other window (see "window" below)

C-x 4 C-o
 kill current and close window (see "window" below)

C-x 4 0
 same as previous C-u M-x quit-window
 list all (see "buffer menu" below) in other window

C-x C-b
 list in current window M-x buffer-menu
 list in other window and select it

M-x buffer-menu-other-window
 avoid switching to current buffer and switch to last

M-x bury-buffer
 undo last bury M-x unbury-buffer
 switch to previously switched buffer

C-x b M-p M-p RET
 bury buffer and switch to next buffer C-x C-<right>
 switch to previously buried buffer C-x C-<left>

C-x C-s
 save current contents to file on disk C-x C-s
 rename current to NAME M-x rename-buffer RET <NAME>
 remove "iX_i" suffix from buffer name if possible

M-x rename-uniquely
 restore contents with file on disk (see "undo" below)

M-x revert-buffer
 same as previous C-x C-v RET

M-x insert-buffer
 copy all of another to point in current

M-x insert-buffer
 copy region to other (see "region" below)

M-x append-to-buffer
 overwrite BUFFER with region (see "region" below)

M-x copy-to-buffer
 permanently enable

M-x enable-command RET erase-buffer RET
 delete everything (see "delete" below)

M-x erase-buffer
 open an indirect buffer based on current

M-x clone-indirect-buffer
 open an indirect buffer but in another window

compare two functions C-x 4 c
 C-h f car RET C-x o M-x clone-buffer RET C-h f cdr RET
 show differences with color
 M-x highlight-compare-buffers

read only

open FILE as read only C-x C-r <FILE> RET
 toggle write status C-x C-q
 view mode for current buffer M-x view-mode
 view mode for other buffer M-x view-buffer RET RET
 open FILE in view mode M-x view-file <FILE> RET
 view mode for BUFFER

M-x view-buffer RET <BUFFER> RET
 scroll down SPC
 scroll up DEL
 view mode help h
 turn off view mode q
 turn off view mode M-x normal-mode

window

scroll down C-v
 scroll up M-v
 scroll down one line C-1 C-v
 scroll up one line M-1 M-v
 scroll other window down C-M-v
 same as previous C-[C-v
 beginning of buffer M-<
 end of buffer M->
 beginning of other buffer
 M-x beginning-of-buffer-other-window
 end of other buffer M-x end-of-buffer-other-window
 move to first column of center line in display

M-r
 move to first column of first displayed line

M-0 M-r
 move to first column of last displayed line

M-- M-r
 move to first column of fourth displayed line

M-4 M-r
 move to first column of third to last displayed line

C-u - 3 M-r
 split vertically in two C-x 2
 switch between windows C-x o
 switch to a buffer in other window C-x 4 b
 show a buffer in other window C-x 4 C-o
 close current C-x 0
 close all others leaving current C-x 1
 same as previous ESC ESC ESC
 open file in other C-x 4 C-f
 open file in other C-x 4 f
 shrink to fit text C-x -
 equalize window heights C-x +
 list all commands starting with C-x 4 C-h
 enlarge window by 1 line C-x ^
 enlarge window 5 lines C-u 5 C-x ^
 shrink window 5 lines M-- 5 C-x ^

split horizontally C-x 3
 enlarge window 5 columns wider C-u 5 C-x }
 shrink window 5 columns narrower C-5 C-x {
 scroll horizontally right C-x <
 scroll horizontally left C-x >

M-x toggle-truncate-lines
 compare buffers starting at point of both

M-x compare-windows
 continue comparing C-x z
 continue comparing more (see "diff" below)

z
 use color to show all differences
 M-x highlight-compare-buffers

buffer menu

list C-x C-b
 list in current window M-x buffer-menu
 list in other window and select it

M-x buffer-menu-other-window
 list only buffers associated with files C-u C-x C-b
 move down SPC
 move down n
 move down C-n
 move up P
 move up C-p
 toggle current as read only (see "read only" above)

%
 ?
 g
 show modes for current
 update list
 automatically update list every 5 seconds

M-x auto-revert-mode
 toggle list to buffers associated with files

T
 view current in other window (see "window" above)

C-o
 RET
 e
 f
 1
 2
 view current in this window
 go to current in this window
 go to current in only 1 window
 go to current in only 1 window
 open current buffer in View mode (see "read only" above)

V
 b
 m
 C-d
 d
 k
 C-k
 x
 bury current (see "buffer" above)
 mark current and move down
 mark to delete current and move up
 mark to delete current and move down
 mark to delete current and move down
 mark to delete current and move down
 execute marks
 sort by name M-2 M-x Buffer-menu-sort
 sort by size (ascending) M-3 M-x Buffer-menu-sort
 sort by mode M-4 M-x Buffer-menu-sort
 sort by file name M-5 M-x Buffer-menu-sort
 sort by display time M-x Buffer-menu-sort
 quit q

redisplay

with line at center of window C-1
 with current line at top of window C-0 C-1
 with current line at bottom of window C-- C-1
 same as previous M-- C-1
 same as previous C-u -1 C-1
 try to make the top of the current function visible in the window

C-M-1
 same as previous C-[C-1
 with current line at top of window C-M-1 C-M-1

command

execute M-x
 M-x ESC x
 M-x C-[x
 M-x M-x execute-extended-command
 previously executed M-x M-p
 next M-n
 complete name of command (see "completion" above)

M-x TAB
 list completions in other window M-x ?
 list completions starting with "forward"

M-x forward TAB
 list completions containing "line" M-x *line TAB
 search backward M-x M-r <REGEXP>
 search for previously executed M-x M-s <REGEXP>
 show recently typed keys C-h l
 show recently echoed messages C-h e
 repeat last C-x z
 edit and re-evaluate last as Emacs Lisp C-x M-ESC
 same as previous C-x ESC ESC
 same as previous C-x C-[C-[
 same as previous C-x M-:
 ask before running M-x disable-command
 run without asking M-x enable-command
 show recently run commands M-x command-history
 execute command at line in history x

iterative command

repeat next command 4 times C-u
 next command once in opposite direction M--
 next command once in opposite direction C--
 repeat next command 8 times C-u 8
 repeat next command 8 times M-8
 repeat next command 8 times C-8
 repeat next command 8 times C-u 8 C-u
 repeat next command 8 times C-8
 repeat next command 13 times C-u 13
 repeat next command 13 times M-1 M-3
 repeat next command 13 times M-1 3
 repeat next command 13 times C-1 C-3
 repeat next command 13 times C-1 3
 insert 13 ones C-u 13 C-u 1
 repeat next command 3 times in opposite direction
 M-- 3
 repeat next command 3 times in opposite direction
 C-- 3
 repeat next command 3 times in opposite direction

repeat next command 16 times	C-u -3
repeat next command 64 times	C-u C-u
insert 369 zeros	C-u C-u C-u
insert 369 zeros	M-3 M-6 M-9 C-u 0
insert 369 zeros	M-3 6 9 C-u 0

non-iterative command

toggle behavior of next command	C-u
toggle behavior of next command with negative value	M--

macro

start recording	C-x (
same as previous	F3
finish recording	C-x)
same as previous	F4
finish recording and run what was just recorded	C-x e

run last again	C-x e
run last twice	C-x e e
run last 3 times	M-3 C-x e
run last infinite times until error	C-0 C-x e
same as previous	M-0 C-x e
go to each line in region and run last iteratively	C-x C-k r

insert counter and increment	C-x C-k C-i
set counter	C-x C-k C-c
insert numbers 0 through 3 one-per-line	F3 F3 RET F4 F4 F4 F4
same as previous	C-x (C-x C-k C-i RET C-x e e e
insert evens 0 through 6 one-per-line	F3 C-u 2 F3 RET F4 F4 F4 F4

same as previous	C-x (C-2 C-x C-k C-i RET C-x e e e
same as previous	C-x (C-2 C-x C-k C-i RET C-x e e e
insert numbers 1 through 3	C-x C-k C-c 1 RET F3 F3 RET F4 F4 F4 F4

same as previous	C-x C-k C-c 1 RET F3 F3 RET F4 F4 F4 F4
same as previous (see "rectangle" below)	C-x C-k C-c 1 RET C-x (C-x C-k C-i RET C-x e e e

duplicate line 9 times	C-a C-SPC C-3 C-n C-x r N
duplicate line 10 times	C-S-DEL F3 C-y C-9 F4
run last and append more commands	C-S-DEL C-x (C-y C-9 C-x e
same as previous	C-u C-x e
delete current macro from ring	C-u F3
switch to previous macro in ring	C-k C-k C-d
switch to next macro in ring	C-k C-k C-p
edit current	C-k C-k C-n
same as previous	C-x C-k e C-x e
give current NAME	C-x C-k C-e
run NAME	C-x C-k n <NAME>
bind current to C-x C-k 1	M-x <NAME>
bind current to C-x C-k A	C-x C-k b 1
bind current to C-c 1	C-x C-k b A
	C-x C-k b C-c 1

bind current to C-c a	C-x C-k b C-c a
run NAME	M-x <NAME>
edit NAME	C-x C-k e M-x <NAME>
make recently typed keys and edit	C-x C-k e C-h l
same as previous	C-x C-k l
insert key in editor	C-c C-q <KEY>
insert C-x in editor	^X
insert M-x COMMAND in editor	<<COMMAND>>
insert 3 newlines in editor	3 * R E T
same as previous	3 * ^ M
same as previous	3 * <<newline>>
same as previous	3 * L F D
delete line in editor	C-a ;;
same as previous	C-a REM
same as previous	C-a C-k
comment line in editor	C-a ;;;
save edits	C-c C-c
save NAME	

C-x C-f ~/.emacs RET M-x insert-kbd-macro RET <NAME> RET	
save last	
C-x C-f ~/.emacs RET M-x insert-kbd-macro RET RET	
run one step at a time	C-x C-k SPC
run current command	SPC
same as previous	y
run current and consecutive similar commands	

skip command	TAB
skip and delete from macro	f
same as previous	n
same as previous	d
skip current and delete rest of macro	DEL
quit	C-k
same as previous	q
insert one key	C-g
insert series of keys	I
replace current key	i
replace current key with series of keys	R
append to end of macro	r
append after current key	A
	a

key

show recently typed keys	C-h l
set for all buffers	M-x global-set-key
remove binding for all buffers	M-x global-unset-key
define binding for current buffer	

remove binding for current buffer	M-x local-set-key
insert next character literally	M-x local-unset-key
insert literal tab character	C-q
insert literal newline	C-q TAB
insert literal carriage return	C-q C-j
insert literal form feed (page delimiter)	C-q C-m

modify next key with Control	C-q C-l
modify next key with Meta	C-x @ c
modify next key with Shift	C-x @ m
modify next key with Hyper	C-x @ S
	C-x @ h

modify next key with Super	C-x @ s
modify next key with Alt	C-x @ a
list bindings starting with C-x @	C-x @ C-h
modify next key with Control and Meta	
	C-x @ c C-x @ m
modify with Control, Meta and Hyper	
	C-x @ c C-x @ m C-x @ h
list bindings starting with C-x	C-x C-h
list bindings starting with C-c	C-c C-h
list bindings starting with C-x 4	C-x 4 C-h
list bindings starting with C-x 5	C-x 5 C-h
list bindings starting with C-x 8	C-x 8 C-h
list bindings starting with C-h 4	C-h 4 C-h
list bindings starting with C-x M-	C-x ESC C-h
same as previous	C-x C-[C-h
list bindings starting with ESC	
	M-: (describe-bindings "\e") RET
same as previous	
	M-x global-unset-key ESC C-h ESC C-h

list bindings starting with M-ESC	ESC ESC C-h
same as previous	C-[C-[C-h
list bindings starting with M-g	M-g C-h
list bindings starting with M-o	M-o C-h

undo

undo, repeat to further undo	C-x u
undo, repeat to further undo	C-_
undo, repeat to further undo	C-/
undo, then redo	C-/ C-g C-/
undo, undo, then redo, redo	C-/ C-/ C-g C-/ C-/
restore buffer with file on disk	M-x revert-buffer
same as previous	C-x C-v RET
turn off for current buffer	M-x buffer-disable-undo
turn on for current buffer	M-x buffer-enable-undo

search

forward to end of MATCH	C-s <MATCH>
reverse to front of MATCH	C-r <MATCH>
help	C-h k C-s
same as previous	C-s C-h m
show key bindings	C-s C-h b
forward to end of MATCH	C-r C-s <MATCH>
reverse to front of MATCH	C-s C-r <MATCH>
forward to end of second MATCH	C-s <MATCH> C-s
reverse to front of second MATCH	C-r <MATCH> C-r
if not at first match, go to previous match	

	DEL
if at first match, delete character from search string	DEL
always delete character from search string	
	C-M-w
same as previous	C-[C-w
forward to start of second MATCH	

reverse to end of second MATCH	C-s <MATCH> C-s C-r
forward to end of second MATCH	C-r <MATCH> C-r C-s
	C-s <MATCH> C-s C-s DEL
reverse to start of second MATCH	

C-r <MATCH> C-r C-r DEL
 finish search RET
 cancel search if current search is successful

 undo search to last successful search C-g
 search for newline C-g
 search for carriage return C-s C-q C-m
 search for current character C-s C-M-y
 same as previous C-s C-[C-y
 search for next two characters C-s C-M-y C-M-y
 search for current character C-s C-M-y C-M-y DEL
 search for rest of current word C-s C-w
 search for next two words C-s C-w C-w
 search for rest of current word C-s C-w C-w DEL
 search for rest of current line C-s M-s C-e
 undo search for rest of current line

C-s M-s C-e DEL
 search for last killed text (see "kill" below)

C-s C-y
 search for second to last killed text (see "kill" below)

C-s M-y
C-s M-p
C-s M-n
 show previous search
 show oldest stored search
 complete for BEGINNING of stored searches

C-s M-TAB <BEGINNING>
 same as previous C-s C-M-i
 same as previous C-s C-[C-i
 resume last search backward C-s C-s
 resume last search forward C-r C-r
 edit search M-e
 toggle regular expression search (see "regular expression" below)

M-r
M-c
M-s w
 toggle case-sensitivity of search
 toggle word-based search
 search, query, and replace (see "replace" below)

M-%
 interactive query SEARCH and REPLACE

C-s <SEARCH> M-% <REPLACE>
C-u M-%
 query replace by word
 change case-sensitivity of all searches

M-x isearch-toggle-case-fold

non-interactive search

C-s RET
C-r RET
 forward case-sensitive
 backward case-sensitive
 ignoring punctuation and whitespace

M-x word-search-forward
 ignoring punctuation and whitespace

M-x word-search-backward

regular expression

C-M-s
C-[C-s
C-u C-s
C-s M-r
 search forward
 same as previous
 same as previous
 same as previous

C-M-r
C-[C-r
M-r
C-M-s C-s
C-M-r C-s
 search reverse
 same as previous
 toggle off regular expression syntax
 repeat last regular expression forward
 repeat last regular expression backward
 suspend replacement and editing buffer (see "recursive edit" above)

C-r
C-M-c
C-[C-c
C-M-%
 resume query and replace
 same as previous
 regular expression replace
 interactive query replace (see "replace" below)

C-M-s <SEARCH> M-% <REPLACE>
 show matches in buffer (see "occur" below)

M-x occur
M-s o
M-s o
M-x how-many
M-x count-matches
 same as previous
 same as previous
 count matches after point
 same as previous
 delete matching lines to end of buffer

M-x flush-lines
 keep matching lines delete the rest to end of buffer

M-x keep-lines
M-s h r
M-s h l
M-s h p
M-s h u
 highlight all matches in buffer
 highlight all lines matching
 highlight phrase
 un-highlight

replace

M-%
C-M-%
C-[C-%
M-% RET
C-M-% RET
C-s <SEARCH> M-% <REPLACE>
C-M-s <SEARCH> M-% <REPLACE>
 search, query, and replace
 search regular expression, query, and replace

 same as previous
 resume last
 resume last as regular expression
 interactive
 interactive with regular expression

y
SPC
,
n
DEL
^
!
e
above)
 replace one and go to next
 same as previous
 replace but don't move
 skip
 same as previous
 previous
 replace all
 edit replacement
 suspend to edit buffer (see "recursive edit" above)

C-r
C-w
C-M-c
C-[C-c
RET
q
 delete match and suspend to edit buffer
 finish edit and resume
 same as previous
 stop
 stop
 replace "long" for "int" in .c files

C-x d *.c RET Q int RET long

delete

C-d
C-u C-d
C-u 8 C-d
C-u C-u C-d
DEL
M-- C-d
C-u C-u C-u DEL
C-5 DEL
M-1 0 DEL
M-x delete-region
 current character
 next 4 characters
 next 8 characters
 next 16 characters
 character backwards (backspace)
 same as previous
 previous 64 characters
 previous 5 characters
 previous 5 characters
 region
 permanently enable

M-x enable-command RET erase-buffer RET
M-x erase-buffer
 entire buffer
 all lines including current to end of buffer

M-x flush-lines RET RET

kill (cut)

C-SPC C-f C-w
C-1 C-d
C-1 DEL
C-- C-d
M-d
C-k
C-0 C-k
C-S-DEL
C-1 C-k
M-- C-k
C-u C-k
M-k
C-w
M-w
M-d
C-DEL
M-DEL
C-M-k
C-[C-k
C-M-DEL
C-[C-DEL
 character (see "character" below)
 same as previous
 character backwards
 same as previous
 word (see "word" below)
 to end of line (see "line" below)
 beginning of line
 entire line
 line including newline
 to beginning of previous line
 next 4 lines
 sentence (see "sentence" below)
 region (see "region" below)
 region but don't delete (copy)
 word (see "word" below)
 word backwards
 word backwards
 sexp (see "sexp" below)
 same as previous
 sexp backwards
 same as previous
 sentence backwards (see "sentence" below)

C-x DEL

paragraph (see "paragraph" below)

M-x kill-paragraph

paragraph backwards (see "paragraph" below)

M-x backward-kill-paragraph

C-M-w
C-[C-w
C-M-w C-w
 append to next
 same as previous
 region appending to previous
 region appending to previous, but don't delete (copy)

C-M-w M-w
C-M-w C-k
C-M-w M-d
C-M-w M-k
 line appending to previous
 word appending to previous
 sentence appending to previous
 paragraph appending to previous

C-M-w M-x kill-paragraph
C-M-w C-M-k
C-M-w C-M-DEL
M-z
 sexp appending to previous
 sexp backward appending to previous
 delete everything to a character

same as previous M-1 M-z
delete everything to a character backwards

M-- M-z
C-u -1 M-z

yank (paste)

the last kill sequence C-y
the last kill sequence C-y
the 2nd to last kill sequence C-y M-y
same as previous C-2 C-y
replace region with last kill C-w C-y M-y
the last kill sequence with point at beginning C-u C-y

mark

set at current point C-SPC
set at current point C-@
toggle between current point and mark C-x C-x
move to last set mark in current buffer C-u C-SPC
move to last set mark in current buffer C-u C-@
move to last set mark in any buffer C-x C-SPC
move to last set mark in any buffer C-x C-@
buffer C-x h
same as previous M-> M-<
word M-@
paragraph M-h
sentence M-x mark-end-of-sentence
function C-M-h
same as previous C-[C-h
page separated by form feed C-x C-p
sexp (see "sexp" below) C-M-@
same as previous C-[C-@
sexp C-M-SPC
same as previous C-[C-SPC

region

set end-point of region (see "mark" above) C-SPC

set end-point of region C-@
kill (see "kill" above) C-w
kill but don't delete (copy) M-w
count lines, words and characters M-=

M-x enable-command RET narrow-to-region RET
C-x n n
widen C-x n w
delete M-x delete-region
save to FILE M-x write-region RET <FILE>
copy to point in BUFFER

M-x append-to-buffer RET <BUFFER>
save to end of FILE M-x append-to-file RET <FILE>
copy and overwrite BUFFER

M-x copy-to-buffer RET <BUFFER> RET

text register

store region to "a" key C-x r s a
kill region to "a" key C-u C-x r s a

insert region stored to "a" key C-x r i a
store rectangle to "a" key (see "rectangle" below) C-x r r a

point register

store current to "a" key C-x r SPC a
move to point to "a" key C-x r j a

window register

store configuration of windows in frame C-x r w a
restore window configurations C-x r j a

number register

store 1 to "a" key C-u 1 C-x r n a
add 1 to number to "a" key C-u 1 C-x r + a
insert number to "a" key C-x r i a

bookmark register

save default C-x r m RET
save as NAME C-x r m <NAME> RET
move to default C-x r b RET
move to NAME C-x r b <NAME> RET
list C-x r l
save all to file M-x bookmark-save

whitespace

insert space SPC
indent or insert tab (see "indent" below) TAB
C-q TAB
insert literal tab character C-q TAB
insert page separator C-q C-1
insert null C-q 0 RET
remove all whitespace at point except one space M-SPC
remove all whitespace around point M-\
remove whitespace before point C-u M-\
remove whitespace after point C-SPC M-m C-w
remove at end of all lines in buffer M-x delete-trailing-whitespace

remove leading for all lines in region C-a C-0 C-M-\
remove all leading in lines buffer C-x h C-0 C-M-\
display whitespace visually M-x whitespace-mode
configure Whitespace Mode M-x whitespace-toggle-options

indent

line with mode-specific rules TAB
same as previous C-i
line and (for supported modes) expression on subsequent lines C-u TAB
same as previous C-u C-i
go to indentation at beginning of line M-m
center line M-o M-s
region with mode-specific rules C-M-\
same as previous C-[C-\
buffer C-x h C-M-\ C-u C-SPC
buffer C-x h C-M-\ C-u C-SPC
paragraph M-h C-M-\ C-u C-SPC

defun C-M-h C-M-\ C-u C-SPC
page C-x C-p C-M-\ C-u C-SPC
sexp C-M-SPC C-M-\ C-u C-SPC
region to 4 columns C-u C-M-\
line to 4 columns C-SPC C-a C-u C-M-\
buffer to 4 columns C-x h C-u C-M-\
region by one column more C-x TAB
line by one column more C-SPC C-a C-x TAB
same as previous C-a SPC
region by 4 columns more C-u C-x TAB
region by 8 columns more C-u 8 C-x TAB
region by 2 columns less C-u - 2 C-x TAB
insert 4 columns for all lines in buffer

remove from all in lines buffer C-x h C-0 C-M-\
convert spaces in region to tabs M-x tabify
convert tabs in region to spaces M-x untabify
always insert spaces

M-x set-variable RET indent-tabs-mode RET nil RET
indent to tab stop M-i
configure stops M-x edit-tab-stops
insert a stop :
remove a stop SPC
save stops C-c C-c
remove and merge line with previous M-^
indent line to 4th column C-a C-u M-x indent-to

newline

one RET
one C-m
one and indent C-j
one below current and indent C-o
indent current and indent next

M-x reindent-then-newline-and-indent

literal newline C-q C-j
insert literal carriage return C-q C-m
keep text following point at same column C-M-o
same as previous C-[C-o
three C-u 3 RET
three C-u 3 C-m
three and indent C-u 3 C-j
three below current and indent C-u 3 C-o
move text following point at same column without fill prefix C-u C-M-o

line

next C-n
previous C-p
beginning C-a
go to M-g g
go to M-g M-g
in last buffer go to C-u M-g g
in last buffer go to C-u M-g M-g
end C-e
kill to end C-k
kill to beginning C-0 C-k
kill from beginning to end C-a C-k
kill from beginning to end including newline

same as previous C-S-DEL
 kill to end including newline C-a C-k C-k
 kill next 4 C-1 C-k
 kill next 2 C-u C-k
 kill to beginning of previous C-2 C-k
 merge current line with previous M-- C-k
 merge next line with current M-^
 same as previous C-u M-^
 insert new after current and indent (see "newline" above) C-n M-^

 keep text following point at same column (see "newline" above) C-o

 same as previous C-M-o
 when not empty line, remove all empty lines below current C-[C-o

 when only empty line, remove all empty lines C-x C-o

 when empty, remove all but one empty lines C-x C-o

 count lines in region C-x C-o
 count lines in page M-=
 count lines in buffer C-x l
 count lines from point forward C-x h M-= C-u C-SPC

 same as previous C-SPC M-> C-x C-x M-=
 display number M-> C-x C-x M-=
 transpose with previous M-x what-line
 swap with line of mark (see "mark" above) C-x C-t

 alphabetically (see "sort" below) C-0 C-x C-t
 reverse order in region M-x sort-lines
 show lines matches in buffer (see "occur" below) M-x reverse-region

 count matches M-x occur
 delete matches to end of buffer M-x how-many
 delete all lines after point M-x flush-lines

 keep matches to end of buffer, delete the rest M-x flush-lines RET RET

 keep matches in region, delete rest M-x keep-lines

 narrow C-SPC C-SPC C-2 C-SPC M-x keep-lines
 change file to UNIX style line endings C-a C-SPC C-e C-x n n

 change file to DOS C-x RET f unix RET
 change file to Mac C-x RET f dos RET
 change file to UTF-8 C-x RET f utf-8 RET
 show line number in mode line M-x line-number-mode
 change if long lines fold or are truncated M-x toggle-truncate-lines

 change if long lines are wrapped at word boundary M-x toggle-word-wrap

character

forward C-f
 backward C-b
 delete (see "delete" above) C-d
 kill (see "kill" above) C-1 C-d
 transpose C-t
 information of current C-x =
 list properties of current in Help buffer C-u C-x =

 same as previous M-x describe-char
 visit Nth character in buffer M-x goto-char RET <N> RET

 list names of charsets M-x list-characters-sets
 insert null C-q 0 RET
 insert space C-q SPC
 insert space using octal value 40 C-q 40 RET
 use hex for C-q M-x set-variable RET read-quoted-char-radix 16

 insert space using hex value 20 C-q 20 RET
 use decimal for C-q M-x set-variable RET read-quoted-char-radix 10

 insert space using decimal value 32 C-q 32 RET
 insert 8 null characters C-u 8 C-q 0 RET
 insert 8 zeros C-u 8 C-u 0 RET

word
 forward M-f
 same as previous M-<right>
 same as previous C-<right>
 backward M-b
 same as previous M-<left>
 same as previous C-<left>
 kill forward M-d
 kill backward C-DEL
 kill backward M-DEL
 transpose M-t
 mark M-@
 mark next 100 C-u 100 M-@
 mark previous 3 M-- 3 M-@
 count for buffer M-< M-x how-many RET \w+ RET
 same as previous C-x h M-=
 make hyphens word character M-x modify-syntax-entry RET - w RET
 make underscores a word character M-x modify-syntax-entry RET _ w RET

capitalization
 lowercase next word M-l
 lowercase previous word M-- M-l
 lowercase next 4 words C-u M-l
 uppercase next word M-u
 uppercase previous word M-- M-u
 uppercase next 2 C-u 2 M-l
 capitalize next M-c
 capitalize previous M-- M-c
 capitalize next 2 C-u 2 M-c
 permanently enable M-x enable-command RET downcase-region RET
 lowercase region C-x C-l
 permanently enable

M-x enable-command RET upcase-region RET
 uppercase region C-x C-u
 capitalize region M-x capitalize-region

sentence
 beginning M-a
 end M-e
 mark M-x mark-end-of-sentence
 kill forwards to end M-k
 kill backwards to beginning C-x DEL
 transpose M-x transpose-sentences
 fix spaces forward with query and replace M-x repunctuate-sentences

paragraph
 forward M-}
 backward M-{
 mark M-h
 fill M-q
 fill and full justify C-u M-q
 fill all in region M-x fill-region
 fill all region together as one M-x fill-region-as-paragraph

 fill region as separate M-x fill-individual-paragraphs

 set fill column to current point C-x f RET
 set fill column to 72 C-x f 72 RET
 same as previous C-u 72 C-x f
 kill to beginning M-x backward-kill-paragraph
 kill to end M-x kill-paragraph
 kill entire M-{ M-x kill-paragraph
 kill entire M-h C-w
 kill entire but don't delete (copy) M-h M-w
 transpose M-x transpose-paragraphs
 center M-o M-S
 justify region to the left M-x set-justification-left

 justify region to the right M-x set-justification-right

 justify region to full width M-x set-justification-full

 justify region to the center M-x set-justification-center
 unjustify region M-x set-justification-none
 alphabetically M-x sort-paragraphs
 in reverse C-u M-x sort-paragraphs
 expect leading space rather than empty lines M-x paragraph-indent-text-mode

 automatically fill at the end of the line M-x auto-fill-mode

 automatically fill entire paragraph after each edit M-x refill-mode

 distinguish intentional newlines from filling M-x use-hard-newlines

 increase left margin of region by one column C-x TAB

 increase left margin of region by 4 columns

C-u C-x TAB
 increase left margin of region by 8 columns
 C-u 8 C-x TAB
 decrease left margin of region by 2 columns less
 C-u - 2 C-x TAB
 add 4 columns to current paragraph
 M-x increase-left-margin
 remove 4 columns to current paragraph
 M-x decrease-left-margin
 set text to beginning of line as fill prefix
 C-x .
 add the fill prefix in region
 C-M-\
 same as previous
 C-[C-\
 set indentation as fill prefix
 M-m C-x .
 reset fill prefix
 C-a C-x .

page

C-q C-l
 insert separator
 C-x]
 forward
 C-x [
 backward
 C-x l
 count lines
 C-x C-p
 mark
 C-x n p
 narrow
 C-x n w
 widen
 M-x sort-pages
 alphabetically
 M-x what-page
 display number

sexp (parenthetical expressions)

M-(
 insert opening and closing parentheses
 M-)
 reindent current and start new line past closing parenthesis
 C-M-f
 move to the next
 C-[C-f
 same as previous
 C-M-<right>
 merge current line with previous
 C-M-b
 move backward
 C-[C-b
 same as previous
 C-M-<left>
 merge next line with current
 C-M-d
 same as previous
 C-[C-d
 same as previous
 C-M-<down>
 merge previous line with current
 C-M-u
 move up from current and to beginning of outside
 C-[C-u
 same as previous
 C-M-<up>
 move up from current and to end of outside
 M-x up-list
 same as previous
 M-- C-M-u
 kill
 C-M-k
 same as previous
 C-[C-k
 kill backward
 C-M-DEL
 same as previous
 C-[C-DEL
 mark
 C-M-@
 same as previous
 C-[C-@
 transpose
 C-M-t
 same as previous
 C-[C-t
 match all open and closed parentheses in buffer
 M-x check-parens

function

C-M-a
 go to beginning
 C-[C-a
 same as previous
 C-M-e
 go to end
 C-[C-e
 same as previous
 C-M-h
 mark current
 C-[C-h
 same as previous
 C-M-h C-M-\ C-u C-SPC
 indent current
 evaluate current (see "emacs lisp" below)

same as previous
 narrow
 widen

comment

M-;
 new
 M-;
 indent
 C-u M-;
 kill current
 C- M-;
 set end point of region
 C-SPC
 region
 M-x comment-region
 remove comment characters in region
 M-x uncomment-region
 region, or uncomment if already
 C-x C-x M-;
 set to start at point
 C-x ;
 kill on this line
 M-- C-x ;
 insert and align or just align to column on previous line
 C-u C-x ;
 if within one, continue on next line and indent

M-j
 same as previous
 C-M-j
 same as previous
 C-[C-j
 surround region
 M-x comment-box
 merge current line with previous
 M-^
 merge next line with current
 C-u M-^
 same as previous
 C-n M-^

occur

list matching lines for regular expression
 same as previous
 M-x occur
 show matches in region (see "region" above)
 M-s o
 same as previous
 C-x n n M-x occur
 show matches with 3 lines of context
 C-u 3 M-x occur
 show matches with 3 lines before match
 C-u - 3 M-x occur
 help
 ?
 same as previous
 h
 go to current match
 RET
 same as previous
 C-c C-c
 same as previous
 o
 show current match in other window
 C-o
 next match
 n
 previous match
 p
 update list of matching lines
 g
 keep Occur buffer around by cloning
 c

r
 keep Occur buffer around by renaming
 q
 quit
 z
 quit and kill buffer
 e
 edit buffer from Occur buffer

transpose

C-t
 characters (see "character" above)
 M-- C-t
 previous with its previous
 C-u 3 C-t
 forward 3 characters
 C-u C-t
 forward 4 characters
 M-- 3 C-t
 backward 3 characters
 C-u C-t
 backward 4 characters
 M-t
 words (see "word" above)
 C-x C-t
 lines (see "line" above)
 paragraphs (see "paragraph" above)
 M-x transpose-paragraphs
 sentences (see "sentence" above)

M-x transpose-sentences
 parenthetical expressions (see "sexp" above)
 C-M-t

column

M-0 M-x move-to-column
 move to first column
 M-x move-to-column
 move to second column
 M-7 M-x move-to-column
 move to column 8
 show column number in mode line
 M-x column-number-mode
 permanently enable

M-x enable-command RET set-goal-column RET
 move to beginning of center line of window (see "window" above)

M-r
 set column for line motion commands
 C-x C-n
 unset goal column for line motion commands
 C-u C-x C-n

add a ruler to the current buffer's window
 M-x ruler-mode

rectangle

C-x r d
 delete, no kill
 C-x r k
 kill
 C-x r y
 yank
 C-x r c
 convert to whitespace
 C-x r o
 insert whitespace
 C-x r t <STRING>
 replace each line with STRING
 insert STRING at each line
 M-x string-insert-rectangle
 remove leading whitespace
 M-x delete-whitespace-rectangle
 remove leading whitespace
 M-x delete-whitespace-rectangle
 store to register "a"
 C-x r r a
 number region starting at 1
 C-x r N
 number region starting with 010
 C-u C-x r N 10 RET %03d SPC RET

table

insert M-x table-insert
 activate mode at point M-x table-recognize-table
 deactivate mode at point

M-x table-unrecognize-table
 forward cell TAB
 backward cell S-TAB
 inactivate table at point

C-u M-x table-recognize-table
 insert 3 columns C-u 3 M-x table-insert-column
 delete 3 columns C-u 3 M-x table-delete-column
 insert 3 rows C-u 3 M-x table-insert-row
 delete 3 rows C-u 3 M-x table-delete-row
 widen column at point C->
 narrow column at point C-<
 heighten row at point C-}
 shorten row at point C-{
 justify text in cell C-:
 split cell C-|
 merge cell C-c C-c *
 insert sequence of characters or numbers

M-x table-sequence
 convert tab delimited region
 M-x table-capture RET C-q C-i RET C-q C-j RET RET RET
 convert to HTML or LaTeX M-x table-generate-source
 activate mode for all M-x table-recognize
 deactivate mode for all M-x table-unrecognize

delimited text

change settings M-x delimit-columns-customize
 format rectangle (see "rectangle" above)
 M-x delimit-columns-rectangle
 format region M-x delimit-columns-region

align

by guessing separator C-u M-x align
 on entire "section" M-x align-current
 based on regular expression M-x align-regexp
 ask for grouping, number of spaces and repeating

C-u M-x align-regexp

sort

lines alphabetically in region M-x sort-lines
 in reverse C-u M-x sort-lines
 region alphabetically by 3rd word in each line
 C-3 M-x sort-fields
 region numerically by 3rd number in each line
 M-x sort-numeric-fields
 lines alphabetically for column between point and mark

M-x sort-columns
 in reverse C-u M-x sort-columns
 paragraphs alphabetically (see "paragraph" above)
 M-x sort-paragraphs
 in reverse C-u M-x sort-paragraphs
 pages alphabetically (see "page" above)

M-x sort-pages

records alphabetically by key in region

directory (folder)

change working M-x cd
 make PATH including any missing parents

M-x make-directory RET <PATH> RET
 delete NAME M-x delete-directory RET <NAME> RET
 delete NAME M-x copy-directory RET <NAME> RET
 rename current to NAME

M-x rename-file RET RET <NAME> RET
 list current C-x d RET
 same as previous C-x C-f RET
 list current in other window (see "window" above)

C-x 4 d RET
 C-x d .. RET
 C-x C-f .. RET
 C-x C-d RET ^
 C-x C-d RET C-x C-j
 C-x d <PATH> RET
 C-x C-f <PATH> RET
 ?
 ?

show recent errors
 go to listing of current file

M-x dired-jump RET RET
 >
 <
 n
 next file
 P
 j
 g
 s
 ~
 M-x dired-jump
 C-u s

automatically update list every 5 seconds

M-x auto-revert-mode
 search listings (see "search" above) C-s
 open file or directory RET
 same as previous f
 same as previous e
 open in other window o
 display in other window C-o
 show file type y
 add new +
 show listing of current subdirectory i
 show listings of all subdirectories, "fully recursive"

C-u s R RET
 \$
 M-\$
 C-M-n
 C-M-p
 C-x C-f <FILE> RET
 !
 M o+r RET
 O <OWNER> RET
 G <GROUP> RET

move current file to FILE R <FILE> RET
 add ~ to the end of current file's name

R M-n ~RET
 C <FILE> RET
 copy current file to file
 make a tilde copy the current file

C M-n ~RET
 P RET
 P RET M-DEL a2ps
 send current file to default printer
 print current file in Postscript
 send current file to ¡PRINTER¡

P SPC -P SPC <PRINTER>

touch current file T
 compress current file Z
 hardlink current file to FILE H <FILE> RET
 symlink current file to FILE S <FILE> RET
 mark current file m
 unmark current file u
 unmark current file and go to previous DEL
 toggle marks t
 unmark all files U
 same as previous * !
 search marked A
 same as previous M-s a C-s
 search marked with regular expression M-s a C-M-s
 find next match M-,
 kill marked file but don't delete w
 hide marked files k
 copy marked files to another directory C
 move marked files to another directory R
 mark current file for deletion d
 mark files containing PATTERN for deletion

% d <PATTERN>
 mark garbage files for deletion % &
 mark auto-save files for deletion (see "file" above)

 .
 D
 delete numerical backups
 delete marked files
 manually edit listing with WDired

M-x wdired-change-to-wdired-mode
 quit WDired and commit the edits made to the listing

C-c C-c
 C-c ESC
 C-c C-[
 cancel WDired
 same as previous
 list file names matching PATTERN

M-x find-name-dired RET <PATTERN> RET
 list files containing grep PATTERN

M-x find-grep-dired RET <PATTERN> RET
 list files containing REGEXP

M-x find-lisp-find-dired RET <REGEXP> RET
 mark files containing PATTERN % g <PATTERN>
 mark file names matching PATTERN % m <PATTERN>
 same as previous * %
 go to next marked file M-}
 same as previous * C-n
 go to previous marked file M-
 same as previous * C-p
 mark executables * *

mark symlinks	* @
mark directories	* /
quit	q

dired-x

load extra Dired functionality

	M-x load-library RET dired-x RET
mark extension	* .
list parent	C-x C-j
smart shell command	M-!
relative symlink current file to FILE	

	M-x dired-do-relink <FILE> RET
relative symlink current file to FILE	Y <FILE> RET
relative symlink files matching REGEXP to current file	

	%Y <REGEXP> RET
open marked files	F
send file by email	V
hide uninteresting files	M-o
go to subdirectory listing in buffer	M-G

tramp

open FILE in DIR on remote HOST

	C-x C-f /HOST:DIR/FILE
same but use secure copy (SCP)	

	C-x C-f /scp:HOST:DIR/FILE
same but demand the use of SSH	

	C-x C-f /ssh:HOST:DIR/FILE
same but demand version 1 of SSH	

	C-x C-f /ssh1:HOST:DIR/FILE
list contents of DIR on remote HOST	

	C-x C-f /HOST:DIR
same as previous	C-x d /HOST:DIR
start a remote shell at HOST	M-x shell

spell check

word	M-\$
buffer	M-x ispell-buffer
region	M-x ispell-region
words and comments in source file	

	M-x ispell-comments-and-strings
quit	q
resume suspended session	M-x ispell-continue
edit word at point in buffer (see "recursive edit" above)	

	M-r
return to spell check by exiting recursive edit	

	C-M-c
same as previous	C-[C-c
use third suggested choice	4
use first suggested choice	0
quick help	?
continue	SPC
accept for this session	a
add to buffer local dictionary	A
replace word with typed version	r
replace every occurrence of word with typed version	

	R
suspend	X
change default dictionary	

	M-x ispell-change-dictionary
highlight misspellings as you type	

	M-x flyspell-mode
highlight misspellings in source code	

	M-x flyspell-prog-mode
highlight existing misspellings	M-x flyspell-buffer
next misspelling in buffer	C-,
insert correction for current word from dictionary	

	C-.
choose second proposed correction for correct word	

	C-. C-.
propose correction for previous word from dictionary	

	C-;
use second proposed correction for previous word	

	C-; C-;
select correct word at or before point from a pop up menu	

	C-c \$
--	--------

abbreviations

define "cpu" as "computer"

	cpu C-x a i g computer RET
same as previous	computer C-3 C-x a g cpu RET
expand "cpu" to "computer"	cpu C-x a e
same as previous	cpu C-x a '
same as previous	cpu C-x ' '
automatically expand	M-x abbrev-mode
undo expansion	C-/
undo last expand	C-x a u
insert "supercomputer"	super M-' cpu C-x ' ,
define new mode-specific	C-x a i l
list all	C-x a C-l
edit all	C-x a C-e
expand in region	M-x expand-region-abbrevs
save	M-x write-abbrev-file
open	M-x read-abbrev-file
delete all	M-x kill-all-abbrevs

composition

set to French characters	C-\ french-postfix RET
disable input method, subsequent re-enables	

	C-\
help with current input method	C-h C-\ RET
insert a letter E acute	e ' ,
insert a letter E and a quote character	e ' ' ,
insert a letter A grave	a `
insert a letter E grave	e `
insert a letter U grave	u `
insert a letter A circumflex	a ^
insert a letter E circumflex	e ^
insert a letter I circumflex	i ^
insert a letter O circumflex	o ^
insert a letter U circumflex	u ^
insert a letter C with cedilla	c ,
insert a letter C and comma	c , ,

insert a letter E umlaut	e "
insert a letter I umlaut	i "
insert a letter U umlaut	u "
insert an open quotation mark	< <
insert a closed quotation mark	> >
change to Spanish characters	

	C-x RET C-\ spanish-prefix RET
insert a letter I grave	i `
insert a letter O grave	o `
insert a letter N with tilde	n ~
change to German characters	

	C-x RET C-\ german-postfix RET
insert a letter A umlaut	a e
insert the letters A and E, no umlaut	a e e
insert a letter O umlaut	o e
insert the letters O and E, no umlaut	o e e
insert a letter U umlaut	u e
insert the letters U and E, no umlaut	u e e
insert the ligature eszett	s z
insert the letters S and Z	s z z

coding

describe SYSTEM	C-h C <SYSTEM> RET
describe current system	C-h C RET
display coding systems	M-x list-coding-systems

environment

set VAR to VALUE in Emacs	
	M-x setenv <VAR> RET <VALUE> RET
edit current value of VAR	M-x setenv <VAR> RET M-p
unset VAR	C-u M-x setenv <VAR> RET
show value of VAR	M-x getenv <VAR> RET
show current variables	M-x getenv TAB
list running processes in Emacs	M-x list-processes
send SIGNAL to any PROCESS	
	M-x signal-process <PROCESS> RET <SIGNAL>

shell

run command	M-!
insert output of command	C-u M-!
set end point of region	C-SPC
send region to command	M-
replace region with output of command	C-u M-
run command in the background asynchronously	
	M-&
new window	M-x shell
new buffer	C-u M-x shell RET RET
new buffer with NAME.	C-u M-x shell RET <NAME> RET
beginning of command line	C-c C-a
go to command prompt and to the end of command	
	C-c C-e
redisplay buffer so prompt is at bottom (see "redisplay" above)	
	C-c C-e
forward command when looking at multi-command	
	C-c C-f
backwards command when looking at multi-command	
	C-c C-b
kill word backward (see "word" above)	C-c C-w

same as previous M-DEL
 send "end of file" to the process C-c C-d
 send process interrupt C-c C-c
 stop the process C-c C-z
 quit the process C-c C-\
 add newline to command C-c SPC
 same as previous C-q C-j
 list command history C-c C-l
 previous command M-p
 next command M-n
 search history backward M-r
 next history match backward C-c M-r
 history match forward C-c M-s
 show beginning of output for last command

C-c C-r
 save session transcript to FILE C-x C-w <FILE> RET
 write output of last command to FILE

C-c C-s <FILE> RET
 append output of last command to FILE
 C-u C-c C-s <FILE> RET

shell script

start a script named file.sh C-x C-f file.sh RET
 use shell script mode for current buffer

M-x shell-script-mode
 same as previous M-x sh-mode
 specify shell and insert header C-c :
 run the script C-c C-x
 execute region C-M-x
 same as previous C-[C-x
 beginning of command M-a
 end of command M-e
 beginning of function C-M-a
 same as previous C-[C-a
 end of function C-M-e
 same as previous C-[C-e
 indent TAB
 newline and indent C-j
 use indentation level of current line C-c <
 analyze buffer's indentation and show inconsistencies

C-c >
 set indentation level for syntactic type at point

C-c =
 show indentation level at point C-c ?
 insert if statement C-c TAB
 insert for statement C-c C-f
 insert case statement C-c C-c
 insert syntax for temporary file C-c C-t
 insert syntax for function C-c (

executable

set interpreter command and arguments M-x executable-set-magic
 make file a command to display itself
 interpret current file M-x executable-self-display
 go to next error M-x executable-interpret
 find path for COMMAND C-x `

lisp

beginning of defun C-M-a
 same as previous C-[C-a
 end of defun C-M-e
 same as previous C-[C-e
 forward list C-M-n
 same as previous C-[C-n
 backward list C-M-p
 same as previous C-[C-p
 down sexp (see "sexp" above) C-M-d
 same as previous C-[C-d
 upward sexp C-M-u
 same as previous C-[C-u
 upward sexp and forward M-x up-list
 insert parens for sexp M-(

 transpose sexp C-M-t
 mark sexp C-M-SPC
 same as previous C-[C-SPC
 run interpreter C-c C-z
 eval expression at point C-M-x
 same as previous C-[C-x
 insert new comment M-;
 kill current comment C-u M-;
 narrow to current defun C-x n d
 widen C-x n w

emacs lisp

evaluate expression before point C-x C-e
 evaluate expression and insert result at point
 evaluate current defun C-u C-x C-e
 same as previous C-M-x
 prompt for expression then evaluate C-[C-x
 eval expression and insert result at point M-:
 indent (see "indent" above) C-u M-:
 same as previous TAB
 indent line and expression for subsequent lines C-i

same as previous C-u TAB
 indent expression after point C-u C-i
 same as previous C-M-q
 pretty print expression after point C-[C-q
 complete symbol at point C-u C-M-q
 same as previous M-TAB
 same as previous C-M-i
 evaluate expressions in region C-[C-i
 evaluate buffer M-x eval-region
 load FILE M-x eval-buffer
 load current file M-x load-file RET <FILE> RET
 load library M-x load-library
 path of library M-x locate-library
 go to LIBRARY M-x find-library <LIBRARY>
 show all libraries on load-path

M-x locate-library RET TAB
 list definitions in library

M-x apropos-library RET <LIBRARY> RET
 show library conflicts in load-path

M-x list-load-path-shadows
 describe library C-h P
 find packages (see "finder" below) C-h p
 permanently set a variable

M-x customize-set-variable
 temporarily set VAR to VALUE
 M-x set-variable RET <VAR> RET <VALUE> RET
 scope VAR to current buffer

M-x make-local-variable RET <VAR> RET
 make VAR global for current buffer
 M-x kill-local-variable RET <VAR> RET
 byte compile current FILE

M-x byte-compile-file RET <FILE> RET
 same as previous M-x byte-compile-file RET RET
 byte compile every file, recursively
 M-x byte-recompile-directory
 byte compile every file, recursively

M-0 M-x byte-recompile-directory
 byte compile defun M-x compile-defun
 go to source code for compile error (see "compile" below)
 C-x `

go to definition of FUNCTION
 M-x find-function RET <FUNCTION> RET
 go to definition of function at point
 M-x find-function RET RET
 go to definition of VARIABLE
 M-x find-variable RET <VARIABLE> RET
 go to definition of variable at point

M-x find-variable RET RET
 start Emacs Lisp mode if not started
 M-x emacs-lisp-mode
 validate coding style M-x checkdoc
 and spell check comments and documentation strings

M-x checkdoc-ispell
 enable debugger on error M-x toggle-debug-on-error
 enable debugger on quit M-x toggle-debug-on-quit

elisp interaction

evaluate expressions interactively
 M-x lisp-interaction-mode
 evaluate sexp before point and insert results on next line
 C-j
 evaluate current defun (see "emacs lisp" above)
 C-M-x
 same as previous C-[C-x

debug elisp

change whether to start session on error
 M-x toggle-debug-on-error
 change whether C-g starts session
 M-x toggle-debug-on-quit
 debug FUNCTION

M-x debug-on-entry RET <FUNCTION> RET
 help h
 move down SPC
 move down C-n
 move 5 down 5 SPC
 move 2 up 2 SPC
 move up C-p
 backward 2 lines 2 C-p
 go up level in expression TAB
 go down lower level S-TAB
 visit help or source location for thing at point RET

complete evaluation level at current point

c
 quit q
 step into d
 set breakpoint b
 unset breakpoint u
 set breakpoint and continue j
 prompt for return value then continue r
 prompt for expression then evaluate e
 prompt for expression then evaluate and record it R

list functions debugged on entry l
 don't debug FUNCTION

M-x cancel-debug-on-entry <FUNCTION> RET
 don't debug for any function

M-x cancel-debug-on-entry RET RET

source-level debugger

turn on instrumentation for current function definition

M-x edebug-defun
 same as previous C-u C-M-x
 turn off instrumentation for current function definition

C-M-x
 step expression in source SPC
 from any buffer, step into expression in source

C-x X SPC
 t
 step fast T
 stop stepping S
 step to next expression n
 step in i
 step out o
 step forward f
 show last result again in minibuffer r
 set breakpoint b
 unset breakpoint u
 unset breakpoint C-c C-d
 set conditional break on result of EXPRESSION

x <EXPRESSION> RET
 move to next breakpoint B
 continue until next breakpoint g
 continue to next breakpoint B
 continue to breakpoints slowly c
 continue to breakpoints fast C
 stop continuing S

stop debugging and finish G
 visit buffer before running Edebug P
 visit buffer before running Edebug v
 momentarily visit buffer before running Edebug
 move back to current point in source p
 move back to current point in source C-c C-l
 from any buffer, move back to current point in source

C-x X w
 ?
 help
 list bindings starting with C-x X C-x X C-h
 prompt for expression then evaluate e
 show backtrace d
 display frequencies in comments for each line for current function

=
 abort a
 abort C-]
 quit q
 same as previous C-x X q
 quit Q

finder

list keywords C-h p
 describe LIBRARY

M-x finder-commentary RET <LIBRARY> RET
 help ?
 move down n
 move up P
 for keyword at point, list Emacs Lisp libraries

RET
 for package at point, show commentary for Emacs
 RET
 same as previous f
 same as previous SPC
 back to beginning of package directory d
 quit q

C

help C-h m
 version of mode M-x c-version
 show syntax information of current point C-c C-s
 use gnu coding style C-c . gnu RET
 use Kernighan and Ritchie coding style

C-c . k&r RET
 use Linux kernel coding style C-c . linux RET
 use BSD coding style C-c . bsd RET
 add index menu to menu bar for current file

M-x imenu-add-menubar-index
 run compiler (see "compile" below) M-x compile
 rerun compiler M-x recompile
 highlight problematic syntax in current file

M-x cwarn-mode
 highlight problematic syntax for all source files

M-x global-cwarn-mode
 highlight preprocessor conditionals

M-x cpp-highlight-buffer

fold conditional preprocessor blocks

M-x hide-ifdef-mode
 indent TAB
 indent C-i
 go to indentation at beginning of line M-m
 newline and indent C-j
 toggle automatic indentation with "electric" characters

C-c C-l
 toggle automatic newlines C-c C-a
 comment region C-c C-c
 uncomment region C-u C-c C-c
 comment region with 3 asterisks C-u 3 C-c C-c
 comment region C-1 C-c C-c
 remove 1 asterisks from comments in region

M-- C-c C-c
 remove 3 asterisks from comments in region

M-- 3 C-c C-c
 fill comment or string literal M-q
 move to end of current or next word of identifier or field

M-f
 move to beginning of current or next word of identifier or field

M-b
 mark to end of current word or next word of identifier or field

M-@
 toggle whether word motion understands mixed-case identifiers

C-c C-w
 move to end of operator, ident, string, paren, brace

C-M-f
 move to beginning of operator, ident, string, paren, brace

C-M-b
 beginning of statement M-a
 end of statement M-e

C-M-a
 beginning of function definition
 end of function definition C-M-e
 beginning of next function M-- C-M-a
 end of previous function M-- C-M-e

C-x n d
 narrow to function definition
 widen C-x n w

C-M-h
 mark current function
 mark to end of operator, ident, string, paren, brace

C-M-@
 indent lines in region properly C-x C-x TAB
 same as previous C-M-\

C-x h C-M-\ C-u C-SPC
 indent buffer properly
 indent current function properly C-c C-q
 align backslash characters in region C-c C-\

C-u C-c C-\
 delete backslash characters in region
 set backslash column to 48

M-x set-variable RET c-backslash-column 48 RET
 beginning of next preprocessor conditional

C-c C-n
 beginning of previous preprocessor conditional

C-c C-p
 beginning of current preprocessor conditional

end of preprocessor conditional M-- C-c C-u
 hungrily delete all whitespace including newlines backwards

C-c DEL
 same as previous C-c C-DEL
 merge current line with previous M-^
 merge next line with current (see "line" above)

C-u M-^
 same as previous C-n M-^
 hungrily delete all whitespace including newlines forwards

C-c C-d
 always hungrily delete whitespace

M-x c-toggle-hungry-state
 show expansion of macros for region in other window

C-c C-e
 don't indent mid-line

M-x set-variable RET c-tab-always-indent RET nil

compile

M-x compile
 execute a compilation command
 M-x recompile
 execute last compilation command
 C-c C-k
 kill
 C-x '
 go to source code for next error
 M-- C-x '
 go to source code for previous error
 C-x o
 visit compilation buffer
 M-n
 next error
 M-p
 previous error
 M-}
 errors for next file
 M-}
 errors for previous file
 M-}
 go to source code for error specified at point

RET
 same as previous C-c C-c
 recompile g
 prompt for command then recompile C-u g

hide

start Hide Show minor mode if not started already

M-x hs-minor-mode
 toggle current block C-c @ C-c
 hide current block C-c @ C-h
 show current block C-c @ C-s
 hide all C-c @ C-M-h
 show all C-c @ C-M-s
 show blocks should VAR be defined C-c @ d <VAR>
 hide blocks should VAR be undefined C-c @ u <VAR>

tag

index .c and .h files in current directory M-! etags *. [ch]

set index file for current buffer

C-u M-x visit-tags-table
 globally set index file M-x visit-tags-table
 go to definition of symbol in index M-.
 go to definition for a regular expression in index

C-M-.
 go to next definition C-u M-.

go to previous definition M-- M-.
 return back to before you started M-*
 go to entry for regular expression in index

M-x tags-search
 go to next entry in index M- ,
 search and replace for regular expression

M-x tags-query-replace
 complete tag at point M-TAB
 same as previous C-M-i
 same as previous C-[C-i
 complete language symbol, avoid tags, at point

C-u M-TAB
 for a source file M-x list-tags
 lists tags matching PATTERN

M-x tags-apropos RET <PATTERN> RET
 forget all previous M-x tags-reset-tags-tables
 choose tags file to use M-x select-tags-table

make

C-x C-f Makefile RET
 new
 C-x C-f foo.mk RET
 new
 M-n
 next target
 M-p
 previous target
 C-c C-\
 backslash region
 C-c C-c
 comment region
 C-c C-b
 list macros and targets
 C-c C-u
 list up-to-date status for targets

diff

compare OLD file with NEW file

M-x diff RET <OLD> RET <NEW> RET
 compare files but prompt for Diff switches

C-u M-x diff
 compare buffer with file on disk

M-x diff-buffer-with-file
 compare current file with backup on disk

M-x diff-backup
 same as previous M-x diff RET RET
 start Diff Mode if not already started for a file

M-x diff-mode
 go to corresponding location in target (new) file C-c C-c
 go to corresponding location in source (old) file C-c C-c
 C-u C-c C-c
 always go to corresponding location in source file

C-u C-u C-c C-c
 apply current hunk C-c C-a
 revert current hunk C-u C-c C-a
 test current hunk C-c C-t
 test current hunk in reverse C-c C-t
 move start of next hunk M-n
 move to start of previous hunk M-p
 move to start of next file in multiple file patch M-}

M-x change-log-merge RET <FILE> RET
 move to start of previous file in multiple file patch C-x v a

M-
 narrow to hunk C-c C-n
 widen C-x n w
 narrow to file of multiple file patch C-u C-c C-n
 kill the current hunk M-k
 kill the current file in multiple file patch

M-K
 split the hunk in two C-c C-s
 reverse direction of entire patch C-c C-r
 reverse direction of patch in region C-u C-c C-r
 new change log entry using context of current location

C-x 4 a
 convert the entire buffer from unified to context format

C-c C-u
 convert the entire buffer from context to unified format

C-u C-c C-u
 convert the entire buffer
 start ediff session C-c C-e

version control (vc)

C-x v C-h
 list bindings for prefix
 C-x v i
 register file
 check in or out, depending on the current state

C-x v v
 finish editing log for check in
 C-c C-c
 check in or out a specific revision C-u C-x v v
 open past revision in new window C-x v ~
 diff with current revision C-x v =
 diff with specific revision C-u C-x v =
 show log C-x v l
 undo checkout C-x v u
 delete the latest revision C-x v c
 annotate file by each line showing when added and by whom C-x v g
 C-x v d

show checked out files
 tag all the files in directory with NAME

C-x v s RET <NAME> RET
 tag files and create branch

C-u C-x v s RET <NAME> RET
 recursively checkout files for a snapshot

C-x v r <NAME>
 update ChangeLog (see "changelog" below) C-x v a
 merge two revisions C-x v m
 insert revision header keyword C-x v h
 start ediff (see "ediff" below) on file

M-x vc-resolve-conflicts

changelog

start new entry using context of current file

C-x 4 a
 start new entry in current log file
 C-x 4 a
 insert previous log from version control C-c C-p
 fill paragraph following syntax rules M-q
 merge current with log FILE

merge conflict

start Smerge Mode if not started	M-x smerge-mode
move to next	C-c ^ n
move to previous	C-c ^ p
keep base	C-c ^ b
keep mine	C-c ^ m
keep other	C-c ^ o
keep what is under point	C-c ^ RET
keep all	C-c ^ a
combine current with next	C-c ^ c
auto resolve	C-c ^ r
auto resolve entire buffer	M-x smerge-resolve-all

grep

show matches in FILES for REGEXP	
	M-x grep RET <REGEXP> SPC <FILES> RET
show matches in FILES for REGEXP	
	M-x lgrep RET <REGEXP> RET <FILES> RET
show matches in all C files	
	M-x lgrep RET <REGEXP> RET RET
same as previous	M-x lgrep RET <REGEXP> RET ch RET
show matches in C source files	
	M-x lgrep RET <REGEXP> RET c RET
show matches in header files	
	M-x lgrep RET <REGEXP> RET h RET
show matches in ChangeLog files	
	M-x lgrep RET <REGEXP> RET l RET
show matches in Make files	
	M-x lgrep RET <REGEXP> RET m RET
show matches in TeX files	
	M-x lgrep RET <REGEXP> RET tex RET
show matches in HTML files	
	M-x lgrep RET <REGEXP> RET *.html RET
extended regular expressions	
	M-x egrep RET <REGEXP> RET <FILES> RET
case insensitive matching	
	M-x igrep RET <REGEXP> RET <FILES> RET
show matches in entire directory tree	
	M-x grep-find RET <REGEXP> RET
same, but HTML files	
	M-x rgrep RET <REGEXP> RET *.html RET RET
same, but C files	
	M-x rgrep RET <REGEXP> RET RET RET
same, but Emacs Lisp files	
	M-x rgrep RET <REGEXP> RET el RET RET
visit next match	C-x ‘
visit next match	M-g M-n
visit next match	M-g n
visit the previous match	M-- C-x ‘
visit the second next match	C-u 2 C-x ‘
visit the fifth next match	C-u 5 C-x ‘
visit the first match	C-u C-x ‘
visit previous match	M-g p
visit previous match	M-g M-p

move to buffer with listed matches	C-x o
stop	C-c C-k
move to next match	TAB
move to previous match	S-TAB
go to match in file	RET
go to match in file	C-c C-c
show next match in other window	M-n
show next match in other window	n
show previous match in other window	M-p
show previous match in other window	p
scroll down	SPC
scroll up	DEL
move to start of matches in next file	M-}
move to start of matches in next file	}
move to end of matches in previous file	M-{
move to end of matches in previous file	{
automatically show match in other window with follow mode	

disable follow mode	C-c C-f
(see "directory" above)	C-c C-f
	M-x find-grep-dired

locate

show files matching PATTERN	
	M-x locate RET <PATTERN> RET
same, but also match REGEXP	
	M-x locate-with-filter RET <PATTERN> RET <REGEXP>
show	M-x locate-with-filter
next matched file	C-n
previous matched file	C-p
visit current file at	RET
open file in other window	C-o
open current file in dired (see "directory" above)	V

calendar

view previous, current and next months	M-x calendar
help	C-h m
user manual	?
one day forward	C-f
one day backward	C-b
seven days forward	7 C-f
same as previous	C-7 C-f
one week forward	C-n
one week backward	C-p
fifty-two weeks forward	52 C-n
same as previous	C-5 C-2 C-n
one month forward	M-}
one month backward	M-{
twelve months forward	12 M-}
same as previous	M-1 M-2 M-}
one year forward	C-x]
one year backward	C-x [
four years forward	4 C-x]
same as previous	C-u C-x]
beginning of week	C-a
end of week	C-e
beginning of the month	M-a
end of the month	M-e
beginning of year	M-<

end of year	M->
scroll month backward	<
scroll month forward	>
scroll forward 3 months	C-v
scroll backward 3 months	M-v
how many days three months from now	
	C-SPC 3 M-} M-=
go to February 29 of current year	
	g d RET Feb RET 29 RET
go to May	o RET May RET
go to sixtieth day of current year	g D RET 60 RET
go to fifth week of current year	g w RET 5 RET
back to today	.
print day of year	p d
print week of year	p c
holidays on day	h
holidays in next 3 months	a
mark holidays	x
unmark holidays	u
sunrise/sunset	S
lunar phases	M
mark all diary entries (see "diary" later)	
	m
unmark diary entries	u
diary entries for day	d
diary entries for day	d
insert single diary entry	i d
insert weekly entry	i w
insert monthly entry	i m
insert annual entry	i y
insert anniversary	i a
repeat entry every ten days	i c 10 RET
making a two-day diary entry	C-SPC C-f i b
making a week-long diary entry	C-SPC C-n i b
making a two week-long diary entry	C-SPC 2 C-n i b
open diary	s
Export month to an HTML file	H m
Export all months in year to HTML files	H y
Export month to TeX file	t m
Export month in landscape display to TeX file	
	t M
Export week in to TeX file	t w 1
Export year to TeX file	t y
quit	q
ediff	
compare two files	M-x ediff
compare two buffers	M-x ediff-buffers
compare regions by lines in two buffers	
	M-x ediff-regions-linewise
compare regions by word in two buffers	
	M-x ediff-regions-wordwise
compare with backup file	M-x ediff-backup
compare buffer with file	M-x ediff-current-file
compare in version control (see "version control" above)	
	M-x ediff-revision
help	?

toggle vertical and horizontal display	
toggle highlighting mode	h
next difference	n
next difference	SPC
previous difference	p
use a	a
use b	b
undo in a	r a
undo in b	r b
quit	q

browse-url

same as previous	M-x browse-url
same as previous	M-x ffap
visit URL at point in Web browser	

	M-x browse-url-at-point
	M-x browse-url RET RET

same as previous

HTML

help	C-h m
view current file in Web browser	C-c C-v
toggle to view in Web browser on each save	
	C-c C-s
toggle inserting of non-ASCII characters as entities	
	C-c 8
toggle invisibility of tags	C-c TAB
add index menu to menu bar for current file	
	M-x html-imenu-index
turn on XHTML tags	

	M-x set-variable RET sgml-xml-mode RET t RET
--	--

use uppercase tags

M-x set-variable RET skeleton-transformation RET upcase RET	
delete current tag	C-c C-d
delete current tag	C-c DEL
delete next 4 tags	C-u C-c C-d
skip forward tag	C-c C-f
skip forward 5 tags	C-u 5 C-c C-f
skip backward tag	C-c C-f
skip backward 4 tags	C-u C-c C-f
start file with TITLE	C-c C-t html RET <TITLE> RET
same as previous	M-x auto-insert RET <TITLE> RET
insert level one heading	C-c 1
insert level two heading	C-c 2
insert level three heading	C-c 3
insert level four heading	C-c 4
insert level five heading	C-c 5
insert level six heading	C-c 6
insert paragraph tag	C-c RET
close paragraph tag	C-c /
insert line break tag	C-c C-j
insert horizontal rule	C-c C-c -
insert link	C-c C-c h
insert page anchor	C-c C-c n
insert image	C-c C-c i
insert radio buttons	C-c C-c r
insert ordered list	C-c C-c o
insert unordered list	C-c C-c u
insert list item	C-c C-c i

close list item	C-c /
insert bold tag	C-u M-o b
insert italic tag	C-u M-o i
insert bold tag	C-u M-o b
insert underline tag	C-u M-o i
insert attributes to current tag	C-c C-a
insert emphasis tag	C-c C-t em RET
insert strong emphasis tag	C-c C-t strong RET
insert source code tag	C-c C-t code RET
insert definition tag	C-c C-t dfn RET
insert keyboard text tag	C-c C-t kbd RET
insert sample text tag	C-c C-t samp RET
insert variable text tag	C-c C-t var RET
insert preformatted text tag	C-c C-t pre RET
insert span tag for text of CLASS	

	C-c C-t span RET class RET <CLASS> RET
--	--

insert definition list with TERM

	C-c C-t dl RET <TERM> RET RET
--	-------------------------------

insert 1-by-1 table

	C-c C-t table RET h RET d RET DEL RET
--	---------------------------------------

prompt for tag name and possible attributes, then insert

	C-c C-t
--	---------

prompt for tag, and surround next 3 words with tag

	C-u 3 C-c C-t
--	---------------

prompt for tag, and surround next word with tag

	C-1 C-c C-t
--	-------------

prompt for tag, and surround region with tag

	M-- C-c C-t
--	-------------

describe current tag

	C-c ? RET
--	-----------

insert non-breaking space entity

	C-c C-n M-SPC
--	---------------

insert a quoted ampersand

	C-c C-n &
--	-----------

insert a quoted less than character

	C-c C-n <
--	-----------

insert a quoted greater than character

	C-c C-n >
--	-----------

insert comment

	M-;
--	-----

kill comment

	C-u M-;
--	---------

display hierarchy of tags for point

	M-x sgml-show-context
--	-----------------------

check markup with external tool

	M-x sgml-validate
--	-------------------

outline

set major mode	M-x outline-mode
set minor mode	M-x outline-minor-mode
help	C-h f outline-mode
insert heading of current level	C-c RET
show all	C-c C-a
hide all	C-c C-t
show body	C-c C-e
hide body	C-c C-c
show all under current heading	C-c C-s
hide all under current heading	C-c C-d
show headings under current heading	C-c C-i
same as previous	C-c TAB
show all headings under current heading	C-c C-k
hide all headings under current heading	C-c C-q
hide leaves	C-c C-l
hide all others except current	C-c C-o
forward heading	C-c C-n

backward heading	C-c C-p
forward 4 headings	C-u C-c C-n
forward to same level heading	C-c C-f
backward to same level heading	C-c C-b
forward 4 headings at same level	C-u C-c C-f
up to parent heading	C-c C-u
up 4 parent headings	C-u 4 C-c C-u
move current subtree down	C-c C-v
move current subtree up	C-c C-^
mark current subtree	C-c @
increase heading level in current subtree	

	C-c C-<
--	---------

decrease heading level in current subtree

	C-c C->
--	---------

SQL

help	M-x sql-help
open DB2 interactive buffer	M-x sql-db2
open DB2 interactive buffer	M-x sql-informix
open Informix interactive buffer	M-x sql-ingres
open InterBase interactive buffer	M-x sql-interbase
open RELEX Linter interactive buffer	M-x sql-linter
open Microsoft SQL interactive buffer	M-x sql-ms
open MySQL interactive buffer	M-x sql-mysql
open Oracle interactive buffer	M-x sql-oracle
open PostgreSQL interactive buffer	M-x sql-postgres
open SOLID interactive buffer	M-x sql-solid
open SQLite interactive buffer	M-x sql-sqlite
open Sybase interactive buffer	M-x sql-sybase
switch to current interactive	

	M-x sql-product-interactive
--	-----------------------------

turn on SQL mode if not already started

	M-x sql-mode
--	--------------

set text buffer's interactive buffer

	M-x sql-set-sqli-buffer RET *SQL* RET
--	---------------------------------------

send region to interactive buffer

	C-c C-r
--	---------

send paragraph to interactive buffer

	C-c C-c
--	---------

send buffer contents to interactive buffer

	C-c C-b
--	---------

calc

start/stop calc	C-x * c
same as previous	M-x calc
quit	q
start/stop	C-x * *
other window	C-x * o
help for help	h ?
help shown in minibuffer	? ?
all help shown in buffer	h h
tutorial	h t
info manual	h i
command name	h c
command description	h k
command summary	h s
put on stack	RET
same as previous	SPC
swap top stack items	TAB
cycle all stack items	M-TAB

same as previous	C-M-i	fix to 3 digits after floating point	d f 3
same as previous	C-[C-i	3 significant digits and normal notation	M-3 d n
delete from stack	DEL	3 significant digits in scientific notation	
delete all from stack	M-0 DEL		M-3 d s
move point to top of the stack	o	3 significant digits in engineering notation	
scroll right	>		
scroll left	<	toggle grouping digits in sets of 3 with commas	M-3 d e
add	+		
subtract			d g
multiply	*	group in sets of 5	M-5 d g
divide	/	group floating point in sets of 3 with commas	
integer divide	\		M-- d g
raise number to power	^	group floating point in sets of 5	M-- M-5 d g
root	I ^	change display mode without updating stack	
square root	Q		H d
negative	n	manually refresh stack with current display mode	
reciprocal	&		H d SPC
enter algebraic expression	,	enter in binary	2#
re-evaluate	=	enter in octal	8#
fraction mode	m f	enter in hexadecimal	16#
change back to floating point decimal	m f	enter in decimal	10#
enter two-thirds (2/3) as a fraction	' 2 : 3 RET	change display radix	d r
divide 3 from 2 as an "evaluate-to"	' 2 / 3 = > RET	display in binary	d r 2
same as previous	' evalto (2 / 3 RET	same as previous	d 2
show 2/3 as a fraction	m f =	display in octal	d r 8
show 2/3 as a decimal	m f =	same as previous	d 8
convert an integer to a float	f	display in hexadecimal	d r 16
floor a float to an integer	F	same as previous	d 6
round a float to the closest integer	R	display in decimal	d r 10
execute calc named command	x	same as previous	d 0
find hypotenuse of two sides	3 4 RET x hypot	change precision	p
find hypotenuse of two sides	' hypot(3,4)	change to radians mode	m r
find hypotenuse of two sides	3 4 f h	change to degrees mode	m d
algebraic variable or expression	,	change to polar mode	m p
previous stack entry	' \$	change to hours, minutes, and seconds mode	
second previous stack entry	' \$\$		m h
third previous stack entry	' \$\$\$	enter 40 degrees, 11 minutes and 15 seconds	40@11'15"
store top of stack to VAR	s s <VAR>	sine	S
store top of stack to VAR and delete	s t <VAR>	cosine	C
recall VAR to stack	s r <VAR>	tangent	T
unstore VAR to stack	s u <VAR>	convert radians	c r
store top of stack to VAR, evaluate stack, and unstore		convert degrees	c d
	s l <VAR>	convert polar	c p
see list of current variables	s r TAB	convert hms	c h
evaluate variables	=	1 meter	' 1 m
first trail item	t [1 centimeter	' 1 cm
last trail item	t]	1 millimeter	' 1 mm
forward trail item	t n	1 mile	' 1 mi
backward trail item	t p	convert units	u c
current trail item	t h	convert 26.2 miles to kilometers	
yank item from trail to stack	t y		' 26.2 SPC mi RET u c km RET
visit trail	t i	90 km per hour to miles per hour	
leave trail	t o		' 90 SPC km/hr RET u c mph RET
toggle algebra mode	m a	72 degrees Fahrenheit to Celsius	
leading zero notation	d z		' 72 SPC degF RET u c degC RET
normal notation	d n	speed of 26.2 miles in 2:03:38	
scientific notation	d s		
scientific notation	d s		
engineering notation	d e		

```

' hms(26.2) mi RET ' 2@3'38" hr RET /
pace of 26.2 miles in 2:03:38
' 2@3'38" hr RET ' 26.2 mi RET /
same as previous
' hms(26.2) mi RET ' 2@3'38" hr RET / & u r c h
view available units u v
define unit
' 14 SPC lb RET u d st RET Stone (mass) RET
save units permanently to ~/.emacs.d/calc.el

```

timeclock

```

start a project M-x timeclock-in
stop working on the current project

report time to leave M-x timeclock-out

M-x timeclock-when-to-leave-string
visit timelog file M-x timeclock-visit-timelog
reread timelog file after crash M-x timeclock-reread-log
reread timelog file after edited M-x timeclock-reread-log
reread timelog file after restarting emacs M-x timeclock-reread-log

```

games

```

fill a 5-by-5 grid M-x 5x5
find balls in a box M-x blackbox
psychoanalysis M-x doctor
text adventure M-x dunnet
try to get 5 in a row M-x gomoku
multiplication puzzle M-x mpuz
classic video tennis M-x pong
eat dots but not yourself or the walls M-x snake
stack blocks M-x tetris
watch an animation M-x type-break
be told when to take breaks M-x type-break-mode
shut off reminders for breaks M-- M-x type-break-mode

```

animation

```

birthday wishes to NAME M-x animate-birthday-present RET <NAME> RET
strike the drive platter and flip the desired bit M-x butterfly

scramble current text in another buffer M-x dissociated-press

towers of hanoi M-x hanoi
the game of life M-x life
give text in buffer strange capitalization M-x studlify-buffer
give text in region strange capitalization M-x studlify-region
display text tricks M-x zone

```