

C-[is the same as <ESC> key, pressing it once before command acts as if Alt was pressed during command.

C-[C-v is the same as <ESC> C-v and C-M-v (Scroll the bottom window after C-x 2 splits window in two).

C-x d Dired (built-in file manager)

M-x view-mode Toggle view mode (h, then click View, then view-mode to list hotkeys)

C-h k Describe hotkeys and other items

C-h C-h List help hotkeys

C-h a Apropos help about a command

C-h m Display help about current buffer's major mode (C-x 1 to close)

C-h v Describe variable, example: C-h v default-directory

M-x cd Change default-directory (local to current buffer)

g Refresh special buffer (*Ibuffer*, *Help*, *Apropos*, etc.)

C-l Scroll window to make cursor appear at middle, top, bottom position

M-r Put cursor (point) in middle, top, bottom position on the window

M-g M-g Go to line

M-g <TAB> Go to column

C-x C-f Open (find) a file

C-x C-f /sudo:... Open file as root: C-x C-f /sudo:root@localhost:/etc/fstab

C-x C-f /sudo:... Open file as root: C-x C-f /sudo:localhost:/etc/fstab

C-x C-f /sudo:... Open file as other user: C-x C-f /sudo:username@localhost:~username/file.txt

C-x C-f /su:... Open file as other user: C-x C-f /su:username@localhost:~username/file.txt

C-x C-s Save the buffer

C-x s Save all buffers to their files (will ask about each buffer)

C-x C-w Save the buffer as new file

C-x k Close (kill) a buffer

C-x b Switch buffer

C-x C-b Display all open buffers

C-x C-c Exit Emacs

ESC ESC ESC Exit out of prompts, regions, prefix arguments and return to just one window

C-/ Undo changes (also C-_ and C-x u)

C-[C-_ Redo changes (also <ESC> C-_ and C-M-_)

M-x menu-bar-mode Enable/disable menu bar (also M-x men-mod, C-[x men-mod)

F10 Activate the menu bar

C-x 0 Delete the active window

C-x 1 Delete other windows

C-x 2 Split window below

C-x 3 Split window right

C-x o Switch active window

M-x sh-mode Switch major mode to Bash

M-x fundamental-mode Switch major mode to Fundamental

M-x tcl-mode Switch major mode to Tcl

M-x text-mode Switch major mode to Text

M-x auto-fill-mode Automatically break long lines at margin in Text mode

C-x f Set the margin in characters (default is 70) (also C-u 70 C-x f)

M-q Refill the paragraph, or active selection (region)

C-u M-q Refill and justify the paragraph

M-x replace-string	Replace text from cursor (point) to the end of region (also M-x r-st)
C-[<	Move cursor (point) to the top of the buffer. (also <ESC> < and M-< alt-shift-,)
C-[>	Move cursor (point) to the bottom of the buffer. (also <ESC> > and M-> alt-shift-.)
C-x r b	Jump to bookmark
C-x r l	List bookmarks
C-x r m	Set bookmark
C-h v bookmark-default-file	Bookmarks file location (~/.emacs.d/bookmarks)
C-x r <SPC>	Store point in register
C-x r j	Jump to register
C-x r n	Store number in register
C-x r +	Increment number in register
C-u -9 C-x r +	Decrement number in register by 9
C-x r s	Store region in register
C-x r i	Insert contents of register (place point after text)
C-u C-x r i	Insert contents of register (place point before text)
C-<SPC>	Set mark and activate selection (region, movement keys change region); deactivate region
C-g	Deactivate region
C-u C-<SPC>	Return point to last mark position
C-x C-x	Switch point and mark, reactivate last region
M-h	Select current/next paragraph
C-x h	Select whole buffer
C-[C-<SPC>	Select next s-expression (around balanced ", ', [, {, etc.), or word (also C-M-<SPC>)
C-[C-- C-[C-<SPC>	Select s-expression in reverse (also C-M-- C-M-<SPC>)
C-s	Incremental search
C-r	Backward incremental search
C-M-s	Regexp incremental search
C-M-r	Regexp backward incremental search
M-n (inside Isearch)	Next item in search history
M-p (inside Isearch)	Previous item in search history
M-s c (inside Isearch)	Toggle case-sensitivity
M-s r (inside Isearch)	Toggle regexp mode
M-s w (inside Isearch)	Toggle word mode ("hello world" will match "hello-world", "hello_world", etc.)
M-s w	Isearch forward for word
M-s _	Isearch forward for symbol
M-s o (inside Isearch)	Occur mode
M-s o	Occur mode
M-g M-n	Next match in occur, compile, grep
M-g M-p	Previous match in occur, compile, grep
C-[%	Search and replace (also M-%)
C-[% (inside Isearch)	Search and replace (also M-%)
C-[C-%	Regexp search and replace (also C-M-%)
C-[C-% (inside Isearch)	Regexp search and replace (also C-M-%)
C-d	Delete next character
<backspace>	Delete previous character
M-d	Cut (kill) next word

C-<backspace>	Cut (kill) previous word
C-k	Cut (kill) rest of line
C-S-<backspace>	Cut (kill) current line
M-k	Cut (kill) sentence (M-x customize-option sentence-end-double-space)
C-[C-k	Cut (kill) next s-expression (also C-M-k)
C-- C-[C-k	Cut (kill) previous s-expression (also C-- C-M-k)
M-z	Cut (kill) to character (including that character)
C-[C-z	Cut (kill) to character (not including, not bound, also C-M-z)
C-w	Cut (kill) active region
M-w	Copy active region
C-[C-w	Next cut command will append to previous cut (also C-M-w)
M-y	Cycle through clipboard (kill ring)
C-y	Paste (yank) last cut/copy
C-t	Transpose characters
M-t	Transpose words
C-x C-t	Transpose lines
C-x C-;	Comment/Uncomment line
M-j	Insert newline with comment (if point is on commented line)
M-;	Comment empty line; add comment at end of line; un/comment selection
C-x C-u	Uppercase selection (region)
C-x C-l	Lowercase selection (region)
M-x upcase-initials-region	Capitalize selection (region)
M-u	Uppercase next word
M-l	Lowercase next word
M-c	Capitalize next word
C-u -2 C-[u	Uppercase 2 previous words (also M-- M-2 M-u and C-- C-2 M-u)
M-x delete-duplicate-lines	Delete duplicate lines (read C-h x delete-duplicate-lines)
M-x flush-lines	Delete (flushes) all lines in selection that match regexp
M-x keep-lines	Delete all lines in selection that do not match regexp
M-x kill-matching-lines	Cut (kill) all lines that match regexp
M-x copy-matching-lines	Copy all lines that match regexp
C-o	Insert blank line after cursor (point), does not move cursor
C-x C-o	Delete all blank lines after cursor (point)
C-[C-o	Split line after cursor (point), keep indentation (also C-M-o)
C-[^	Join line with cursor (point) and previous line (also M-^)
M-x delete-trailing-whitespace	Delete all trailing whitespace in buffer
C-[:, (setq show-trailing-whitespace t)	Show trailing whitespace in current buffer
C-[:, (setq-default show-trailing-whitespace t)	Show trailing whitespace in all buffers
M-<SPC>	Delete all but 1 space/tab around cursor (point)
M-\	Delete all spaces/tabs around cursor (point)
M-x whitespace-mode	Toggle space/tab/newline highlighting
M-/	Auto-complete word
C-[C-/	Display list of auto-completions from current buffer (also C-M-/)
<TAB>	Indent line/selection
C-x <TAB>	Indent selection by number if has argument, or interactive

C-u 20 C-x <TAB>	Indent selection (region) by 20 columns
C-[\$	Spell check the word at cursor (point) (also M-\$)
C-.	Correct word at cursor (if flyspell-mode, or flyspell-prog-mode)
C-q	Insert control character (example: C-q C-l)
M-s h p	Highlight phrase
M-s h r	Highlight regexp
M-s h .	Highlight symbol at cursor (point)
M-s h u	Remove highlighting
M-x display-fill	Toggle display of fill column (M-x display-fill-column-indicator-mode)
M-x hl-line-mode	Highlight line at cursor (color inherited from highlight)
M-x list-faces-display	List all currently loaded faces
M-x list-colors-display	List all generic colors

C-[:	Evaluate expression, like (default-value 'fido-mode), (also M-:)
M-x eval-expression	Evaluate (default-value 'tab-width) - value in current buffer
C-x C-[C-[Re-evaluate last command (also C-x <ESC> <ESC> and C-x M-<ESC>)
C-[: (customize-face)	Customize fonts and colors ('Font Lock' categories)
C-u M-x what-cursor-position	Describe character at point
C-x 8 <RET> 401	Input character #x401 (puts \u0401), M-x fido-mode to disable Fido
M-x shell	Invoke Unix shell (bash, dash, etc., also works through TRAMP)
M-x eshell	Invoke shell implemented in elisp
M-x ansi-term	Invoke terminal emulator
M-x proced	Top/htop functionality in Emacs
M-x man	Read man pages in Emacs
C-[!	Invoke shell command, print output (also M-!)
C-u C-[!	Invoke shell command, insert into current buffer (also C-u M-!)
C-[Pipe selection (region) to shell command, print output (M-)
C-u C-[Pipe selection to shell command, replace selection (C-u M-)
M-x compile	Run command (shell script, program, etc.), g to re-run command
M-x customize-themes	*Custom Themes* buffer (enable theme, change/migrate settings)
M-x customize-create-theme	Create a custom theme
M-x variable-pitch-mode	Toggle fixed-pitch/variable-pitch font in current buffer
M-x buffer-face-mode	Toggle fixed-pitch/variable-pitch font and other font modes

Fido mode:	M-x customize-mode, incomplete-mode
C-s	Move to next match
C-r	Move to previous match
<TAB>	Traditional non-Fido TAB-completion
<RET>	Open selected match
C-d	Open dired in current directory
//	Go to root directory /
~/	Go to home directory ~
<backspace>	Delete a character or go up one directory
M-j	Stop autocomplete after typing file name, choose what was typed

Dired hotkeys:

C-x d	Open Dired file manager
C-x 4 d	Open Dired in other window
M-x dired	Open Dired file manager
C-x C-f C-d	Open Dired if fido-mode is enabled
M-x customize-group, dired	Customize Dired settings, write changes to ~/.emacs.d/init.el
C-h m (in Dired buffer)	Open help about Dired, list all Dired hotkeys
q	Quit Dired
<RET>	Open (visit) current item
f	Open (visit) current item
F	Open (visit) all marked items, (require 'dired-x)
C-u F	Open all marked items in background, (require 'dired-x)
^	Go up one directory
p	Move cursor (point) up
C-p	Move cursor (point) up
n	Move cursor (point) down
C-n	Move cursor (point) down
m	Mark current item
u	Unmark current item
U	Unmark all items
d	Flag for deletion
* m	Mark selection
* u	Unmark selection
* %	Mark by elisp regexp, example .*\.tcl
* .	Mark by extension
t	Toggle marking for all items
* t	Toggle marking for all items
* c	Change mark type of all items with same mark type
C	Copy items
R	Rename, or move items
O	Change owner
G	Change group
M	Change permissions
D	Delete marked items (files with * mark)
X	Delete flagged items (files with D mark)
c	Compress marked into an archive file
g	Refresh Dired buffer
+	Create a subdirectory
s	Toggle sorting by name/date
<	Move cursor (point) to previous subdirectory
>	Move cursor (point) to next subdirectory
j	Move cursor (point) to file, prompt for filename
M-s a C-s	Begin multifile Isearch, (require 'dired-x)
Q	Xref Query replace regexp, (require 'dired-x)
A	Xref Search by regexp, (require 'dired-x)

!	Synchronous shell command, *, ?, (require 'dired-x)
&	Asynchronous shell command, (require 'dired-x)
i	Insert subdirectory in current Dired buffer
\$	Collapse/expand directory opened in Dired buffer
M-x find-dired	Call find shell command with pattern
M-x find-name-dired	Call find shell command with -name
M-x find-grep-dired	Call find with grep
M-x find-lisp-find-dired	Call elisp find implementation, elisp regexp