GNU Emacs Survival Card

for version 31

In the following, `C-z` means hit the '`z`' key while holding down the *Ctrl* key. `M-z` means hit the '`z`' key while hitting the *Meta* (labeled *Alt* on some keyboards) or after hitting *Esc* key.

Running Emacs

To enter GNU Emacs, just type its name: `emacs`. Emacs divides the frame into several areas: menu line, buffer area with the edited text, mode line describing the buffer in the window above it, and a minibuffer/echo area in the last line.

`C-x C-c`	quit Emacs
`C-x C-f`	edit file; this command uses the minibuffer to read the file name; use this to create new files by entering the name of the new file
`C-x C-s`	save the file
`C-x k`	kill a buffer
`C-g`	in most context: cancel, stop, abort partially typed or executing command
`C-x u`	undo

Moving About

`C-l`	scroll current line to center of window
`C-x b`	switch to another buffer
`M-<`	move to beginning of buffer
`M->`	move to end of buffer
`M-g M-g`	go to a given line number

Multiple Windows

`C-x 0`	remove the current window from the display
`C-x 1`	make active window the only window
`C-x 2`	split window horizontally
`C-x 3`	split window vertically
`C-x o`	move to other window

Regions

Emacs defines a 'region' as the space between the *mark* and the *point*. A mark is set with `C-`*space*. The point is at the cursor position.

`M-h`	mark entire paragraph
`C-x h`	mark entire buffer

Killing and Copying

`C-w`	kill region
`M-w`	copy region to kill-ring
`C-k`	kill from the cursor all the way to the end of the line
`M-DEL`	kill word
`C-y`	yank back the last kill (`C-w C-y` combination could be used to move text around)
`M-y`	replace last yank with previous kill

Searching

`C-s`	search for a string
`C-r`	search for a string backwards
`RET`	quit searching
`M-C-s`	regular expression search
`M-C-r`	reverse regular expression search

Use `C-s` or `C-r` again to repeat the search in either direction.

Tags

Tags tables files record locations of function and procedure definitions, global variables, data types and anything else convenient. To create a tags table file, type '`etags` *input_files*' as a shell command.

`M-.`	find a definition
`M-,`	pop back to where `M-.` was last invoked
`M-x tags-query-replace`	run query-replace on all files recorded in tags table

Compiling

`M-x compile`	compile code in active window
`C-c C-c`	go to the next compiler error, when in the compile window or
`C-x '`	when in the window with source code

Dired, the Directory Editor

`C-x d`	invoke Dired
`d`	flag this file for deletion
`~`	flag all backup files for deletion
`u`	remove deletion flag
`x`	delete the files flagged for deletion
`C`	copy file
`g`	update the Dired buffer
`f`	visit the file described on the current line
`s`	switch between alphabetical date/time order

Reading and Sending Mail

`M-x rmail`	start reading mail
`q`	quit reading mail
`h`	show headers
`d`	mark the current message for deletion
`x`	remove all messages marked for deletion
`C-x m`	begin composing a message
`C-c C-c`	send the message and switch to another buffer
`C-c C-f C-c`	move to the 'Cc' header field, creating one if there is none

Miscellaneous

`M-q`	fill paragraph
`M-/`	expand previous word dynamically
`C-z`	iconify (suspend) Emacs when running it under X or shell, respectively
`M-x revert-buffer`	replace the text being edited with the text of the file on disk

Query Replace

`M-%`	interactively search and replace
`M-C-%`	using regular expressions

Valid responses in query-replace mode are

`SPC`	replace this one, go on to next
`,`	replace this one, don't move
`DEL`	skip to next without replacing
`!`	replace all remaining matches
`^`	back up to the previous match
`RET`	exit query-replace
`C-r`	enter recursive edit (`M-C-c` to exit)

Regular Expressions

`.` (dot)	any single character except a newline
`*`	zero or more repeats
`+`	one or more repeats
`?`	zero or one repeat
`[...]`	denotes a class of character to match
`[^...]`	negates the class
`\`*c*	quote characters otherwise having a special meaning in regular expressions
`...|...|...`	matches one of the alternatives ("or")
`\( ... \)`	groups a series of pattern elements to a single element
`\`*n*	same text as *n*th group
`^`	matches at line beginning
`$`	matches at line end
`\w`	matches word-syntax character
`\W`	matches non-word-syntax character
`\<`	matches at word beginning
`\>`	matches at word end
`\b`	matches at word break
`\B`	matches at non-word break

Registers

`C-x r s`	save region in register
`C-x r i`	insert register contents into buffer
`C-x r SPC`	save value of point in register
`C-x r j`	jump to point saved in register

Rectangles

`C-x r r`	copy rectangle to register
`C-x r k`	kill rectangle
`C-x r y`	yank rectangle
`C-x r t`	prefix each line with a string
`C-x r o`	open rectangle, shifting text right
`C-x r c`	blank out rectangle

Shells

`M-x shell`	start a shell within Emacs
`M-!`	execute a shell command
`M-|`	run a shell command on the region
`C-u M-|`	filter region through a shell command

Spelling Check

`M-$`	check spelling of word at the cursor
`M-x ispell-region`	check spelling of all words in region
`M-x ispell-buffer`	check spelling of entire buffer

International Character Sets

`C-x RET C-\`	select and activate input method for the current buffer
`C-\`	enable or disable input method
`M-x list-input-methods`	show all input methods
`M-x set-language-environment`	specify principal language
`C-x RET c`	set coding system for next command
`M-x find-file-literally`	visit file with no conversion of any kind
`M-x list-coding-systems`	show all coding systems
`M-x prefer-coding-system`	choose preferred coding system

Keyboard Macros

`C-x (`	start defining a keyboard macro
`C-x )`	end keyboard macro definition
`C-x e`	execute last-defined keyboard macro
`C-u C-x (`	append to last keyboard macro
`M-x name-last-kbd-macro`	name last keyboard macro

Simple Customization

`M-x customize`	customize variables and faces

Getting Help

Emacs does command completion for you. Typing `M-x` *tab* or *space* gives a list of Emacs commands.

`C-h`	Emacs help
`C-h t`	run the Emacs tutorial
`C-h i`	enter Info, the documentation browser
`C-h a`	show commands matching a string (apropos)
`C-h k`	display documentation of the function invoked by keystroke

Emacs gets into different *modes*, each of which customizes Emacs for editing text of a particular sort. The mode line contains names of the current modes, in parentheses.

`C-h m`	get mode-specific information

Copyright © 2026 Free Software Foundation, Inc.
For GNU Emacs version 31
Author Włodek Bzyl (matwb@univ.gda.pl)
Released under the terms of the GNU
General Public License version 3 or later.
For more Emacs documentation, and the TEX
source for this card, see the Emacs distribution,
or `https://www.gnu.org/software/emacs`