

# The (Nearly) Complete GNU Emacs Key Bindings (plus Epoch)

## Moving Around in Text

Key	Binding
<b>c-v</b>	Forward one screen
<b>a-v</b>	Back one screen
<b>c-l</b>	Clear and redisplay
<b>c-p</b>	Previous line
<b>c-n</b>	Next line
<b>c-b</b>	Backward one character
<b>c-f</b>	Forward one character
<b>a-b</b>	Backward one word
<b>a-f</b>	Forward one word
<b>c-a</b>	Beginning of line
<b>c-e</b>	End of line
<b>a-a</b>	Beginning of sentence
<b>a-e</b>	End of sentence
<b>a-&lt;</b>	Beginning of file
<b>a-&gt;</b>	End of file
<b>&lt;arg&gt; c-l</b>	Scroll to line “arg”
<b>c-x &lt;</b>	Scroll text left
<b>c-x &gt;</b>	Scroll text right
<b>c-x \$</b>	Show deeply indented lines
<b>&lt;arg&gt; c-x \$</b>	Hide lines over “arg” columns indented

## Deleting, Undo, and Yank

Key	Binding
<b>c-x u</b>	Undo
<b>c-_</b>	Undo
<b>&lt;bksp&gt;</b>	Delete char before cursor
<b>&lt;del&gt;</b>	Delete char after cursor
<b>c-d</b>	Delete char after cursor
<b>a-&lt;bksp&gt;</b>	Kill word before cursor
<b>a-d</b>	Kill word after cursor
<b>c-k</b>	Kill to end of line
<b>a-k</b>	Kill to end of sentence
<b>c-x &lt;del&gt;</b>	Kill to beginning of sentence
<b>a-\</b>	Delete tabs and spaces around point

<b>a-&lt;space&gt;</b>	Delete tabs and spaces, leave one space
<b>c-y</b>	Yank back from kill buffer
<b>a-y</b>	Meta-yank (yank previous)
<b>c-a-w</b>	Append next kill to last killed text
<b>c-x c-o</b>	Delete all but one (of many) blank lines

## Files and Buffers

Key	Binding
<b>c-x c-f</b>	Find (read) a file
<b>c-x c-v</b>	Visit a different file
<b>c-x c-s</b>	Save a file
<b>c-x c-w</b>	Save buffer is specified file.
<b>a-x set-visited-file-name</b>	Change filename for this buffer
<b>c-x c-c</b>	Quit Emacs
<b>a-x auto-save-mode</b>	Toggle or set auto-save mode
<b>a-x recover-file RET &lt;name&gt; RET; c-x c-s</b>	Recover auto-saved file
<b>c-x s</b>	Save some buffers
<b>c-x c-b</b>	List buffers
<b>c-x k</b>	Kill a buffer
<b>a-x kill-some-buffers</b>	Offer to kill each buffer
<b>a-~</b>	Forget that current buffer is changed.
<b>c-x 1</b>	1 screen
<b>c-x 2</b>	2 screens
<b>c-z</b>	Exit temporarily
<b>c-x c-d &lt;dir&gt;</b>	Short directory listing
<b>c-u c-x c-d &lt;dir&gt;</b>	Long directory listing
<b>c-x d or a-x dired</b>	Enter directory editor (dired)
<b>c-x 4 d</b>	Enter dired in another window
<b>a-x view-file</b>	View a file
<b>a-x insert-file</b>	Get file into current buffer
<b>a-x write-region</b>	Write region into file
<b>a-x delete-file</b>	Delete file
<b>a-x rename-file</b>	Rename file

<b>a-x add-name-to-file</b>	Add another name to this file (link)
<b>a-x copy-file</b>	Copy the file
<b>a-x make-symbolic-link</b>	Link two files
<b>a-x pwd</b>	Print pwd
<b>a-x cd</b>	Change pwd

## In Buffer List (c-x c-b)

Key	Binding
<b>d or k</b>	Delete (kill) the buffer
<b>s</b>	Request save of buffer
<b>~</b>	Mark buffer as unmodified
<b>x</b>	Perform saves and deletions
<b>1</b>	Select buffer in full screen window
<b>2</b>	Select buffer in other of 2 windows
<b>f</b>	Select buffer to edit (full screen)
<b>o</b>	Select buffer in other window
<b>m</b>	Flag for display when “q” is used
<b>q</b>	Display all flagged buffers

## In Dired List (c-x d)

Key	Binding
<b>d</b>	Flag file for deletion
<b>u</b>	Remove deletion flag
<b>x</b>	Delete files flagged
<b>#</b>	Flag all auto-save files for deletion
<b>~</b>	Flag all backup files for deletion
<b>.</b>	Flag excess backup files for deletion
<b>c</b>	Copies file
<b>f</b>	Visits (edit) file
<b>o</b>	Like f, but use another window
<b>r</b>	Rename file
<b>v</b>	View the file (read-only editing)

## In View File (a-x view-file)

Key	Binding
<b>&lt;space&gt;</b>	Move forward one window

<b>&lt;del&gt;</b>	Move backward one window
<b>c-c</b>	Exit

<b>c-h w &lt;cmd&gt;</b>	Show keys which run “cmd”
--------------------------	---------------------------

<b>a-x insert-buffer</b>	Insert region into buffer at point
<b>a-x append-to-file &lt;file&gt;</b>	Append region to file

## Epoch Additions

In the following commands, a “window” refers to an X-window, not an Emacs window.

Key	Binding
<b>c-z 4 c-f</b>	Find (read) a file in another window.
<b>c-z 4 b</b>	Select a buffer in another window.
<b>c-z 0</b>	Delete current window.
<b>c-z o</b>	Circulate through mapped windows in forward direction.
<b>c-z p</b>	Circulate through mapped windows in backward direction.
<b>c-z 2</b>	Duplicate current window.
<b>c-z h</b>	Hide (unmap) current window.
<b>c-z r</b>	Raise current window.
<b>c-z l</b>	Lower current window.
<b>c-z m</b>	Raise minibuffer window.
<b>c-z i</b>	Iconify current window.
<b>c-z a</b>	Force update of title bar for current window.

## Help

Key	Binding
<b>c-h</b>	Help
<b>c-h a &lt;keyword&gt;</b>	List all commands which contain “keyword”
<b>c-h b</b>	Display all key bindings
<b>c-h c &lt;cmd&gt;</b>	What does “cmd” do?
<b>c-h f &lt;function&gt;</b>	Help for function
<b>c-h i</b>	Run Info (for browsing docs)
<b>c-h k &lt;cmd&gt;</b>	More extensive help
<b>c-h l</b>	Description of last 100 characters typed
<b>c-h m</b>	Docs for current major mode
<b>c-h n</b>	Docs for Emacs changes
<b>c-h t</b>	Tutorial

## Arguments and Re-executing

Key	Binding
<b>c-u &lt;arg&gt;</b>	Numeric argument to next command
<b>a-&lt;digit 1..9&gt;</b>	Numeric argument to next command
<b>c-g</b>	Bail-out
<b>c-x ESC</b>	Re-execute mini-buffer command
Key	Binding
<b>a-p</b>	Move to previous command
<b>a-n</b>	Move to next command
<b>a-x list-command-history</b>	Display entire command history

## Mark, Point, and Region

Key	Binding
<b>c-&lt;space&gt; or c-@</b>	Set mark
<b>c-x c-x</b>	Swap mark and point
<b>a-@</b>	Set mark after end of next word
<b>a-h</b>	Set region around current paragraph
<b>c-x h</b>	Set region around entire buffer
<b>c-x c-p</b>	Set region around current page

## Region Commands

Key	Binding
<b>c-u c-&lt;space&gt;</b>	Move through locations in mark ring
<b>c-x c-u</b>	Uppercase text in region
<b>c-x c-l</b>	Lowercase text in region
<b>c-w</b>	Kill a region
<b>a-w</b>	Save region as killed, without killing
<b>c-x x &lt;arg&gt;</b>	Save region in register “arg”
<b>a-g</b>	Fill each paragraph in region
<b>a-x fill-region-as-paragraph</b>	Fill entire region as one paragraph
<b>a-x print-region</b>	Print a region
<b>c-x TAB or c-a-\</b>	Indent a region
<b>c-x a &lt;buffer&gt;</b>	Append region to buffer
<b>a-x prepend-to-buffer</b>	Prepend region to buffer
<b>a-x copy-to-buffer</b>	Copy region into buffer (deletes old contents)

## Rectangles

Key	Binding
<b>a-x delete-rectangle</b>	Delete region and close
<b>a-x kill-rectangle</b>	Delete and put in kill buffer
<b>a-x yank-rectangle</b>	Yank last killed rectangle
<b>a-x open-rectangle</b>	Insert blank space to fill region
<b>a-x clear-rectangle</b>	Convert region to blanks

## Formatting

Key	Binding
<b>c-o</b>	Insert blank lines after cursor
<b>a-l</b>	Lowercase following word
<b>a-u</b>	Uppercase following word
<b>a-c</b>	Capitalize following word
<b>a- a-l</b>	Convert last word to lower case
<b>a- a-u</b>	Convert last word to upper case
<b>a- a-c</b>	Convert last word to capitalized
<b>c-u &lt;arg&gt; c-x f</b>	Set margin to “arg”
<b>a-q</b>	Re-fill paragraph
<b>a-x auto-fill-mode</b>	Toggle auto-fill
<b>c-u &lt;arg&gt; a-q</b>	Justify text
<b>a-s</b>	Center line
<b>c-t</b>	Transpose two characters
<b>a-t</b>	Transpose two words
<b>c-x c-t</b>	Transpose two lines

## Search and Replace

Key	Binding
<b>c-s</b>	Search
<b>c-r</b>	Reverse search
<b>c-s ESC &lt;string&gt; RET</b>	Non-incremental search
<b>c-r ESC &lt;string&gt; RET</b>	Non-incremental reverse search
<b>c-s ESC c-w &lt;words&gt; RET</b>	Search for “words,” ignore punctuation
<b>c-r ESC c-w &lt;words&gt; RET</b>	Search for “words,” ignore punctuation

**a-x replace-string** **RET** <string> **RET** <newstring>  
 Replace all occurrences of “string”

**a-%** <string> **RET** <newstring> Query replace “string”

**a-x query-replace** <string> **RET** <newstring> Query replace “string”

**a-x occur** Show each line containing “regexp”

**a-x list-matching-lines** Show each line containing “regexp”

**c-a-s** Incremental search for “regexp”

**a-x re-search-forward** Non-incremental search for “regexp”

**a-x re-search-backward** Non-incremental search for “regexp”

**a-x count-matches** Print number of matches following point for “regexp”

**a-x delete-non-matching-lines** Delete each line following point that doesn't contain “regexp”

**a-x delete-matching-lines** Delete each line following point that contains “regexp”

## Query Replace (a-x query-replace)

Key	Binding
<space>	Replace “string” with “newstring”
<del>	Skip this occurrence
<ESC>	Exit without any more changes
<.>	Change this one and then exit
<!>	Change all remaining
<^>	Back-up one occurrence
c-h	Show your options

## Regular Expressions

Expression	Meaning
.	Any char except \n.
*	Repeat preceding as many times as possible. “fo*” matches “f” followed by any number of “o”s. Zero “o”s matches.
+	Like *, but one occurrence is necessary.
?	Like *, but must match one or zero times.

[...]	Contains a character set. [a–z] matches all lower-case letters. [ad] matches one “a” or “d.” [ad]* matches any string of “a”s and “d”s.
[^...]	Contains a complement character set. Matches any character not specified.
^	Matches the empty string (only if at the beginning of a line).
\$	Matches the empty string (only at the end of a line).

## Registers

Key	Binding
<b>a-x view-register</b> <b>RET</b> <arg>	Display contents of register “arg”
<b>c-x /</b> <arg>	Save location of point in register “arg”
<b>c-x j</b> <arg>	Jump to location saved in “arg”
<b>c-x x</b> <arg>	Save region in register “arg”
<b>c-x g</b> <arg>	Insert text from register “arg”
<b>c-x r</b> <arg>	Copy rectangle region to register “arg”
<b>c-x g</b> <arg>	Insert rectangle from register “arg”

## Spell Checking

Key	Binding
a-\$	Check and correct spelling of word
<b>a-x spell-buffer</b>	Check spelling of words in buffer
<b>a-x spell-region</b>	Check spelling of words in region
<b>a-x spell-string</b>	Check spelling of specific word

## Windows

Key	Binding
<b>c-x 2</b>	Split into two vertical windows
<b>c-x 5</b>	Split into two horizontal windows
<b>c-x 0</b>	Select other window
<b>c-a-v</b>	Scroll the next window

<b>a-x compare-windows</b>	Find next place where text doesn't match
<b>c-x 4</b>	Perform command in another window
<b>c-x 0</b>	Get rid of selected window
<b>c-x 1</b>	Get rid of all windows except current
<b>c-x ^</b>	Make current window taller
<b>c-x }</b>	Make current window wider

## Indentation

Key	Binding
<b>TAB</b>	Indent “appropriately” (based on Mode)
<b>c-q TAB</b>	Insert tab character into buffer
<linefeed>	Return followed by Tab
<b>a-m</b>	Move to first non-blank character on line
<b>a-^</b>	Merge two lines
<b>c-a-o</b>	Split line at point (new line is indented to match old line)
<b>c-a-\</b>	Indent several lines to same column
<b>c-x TAB</b>	Shift block of lines rigidly
<b>a-i</b>	Indent to next tab column
<b>a-x indent-relative</b>	Indent to under an indentation point in previous line
<b>a-x edit-tab-stops</b>	Set tab stop columns

## Text Modes

Key	Binding
<b>a-x text-mode</b>	Enter text mode
<b>a-x indented-text-mode</b>	Enter indented text mode
<b>a-x nroff-mode</b>	Enter Nroff mode (filling stops at nroff commands)
	<b>Key Binding</b>
<b>a-n</b>	Move to beginning of next line that isn't an nroff command.
<b>a-p</b>	Move up to next non-command line.

**a-?** Print number of text lines (not command lines) in the region.

**a-x plain-tex-mode** Enter Tex mode.

**a-x latex-mode** Enter LaTeX mode.

## Outline Mode

Lines beginning with a star are considered heading lines. The number of stars determines the depth of the heading. Lines without a star are body lines. Body lines belong to the preceding heading line.

Key	Binding
<b>a-x outline-mode</b>	Enter outline mode
<b>c-c c-n</b>	Move to next visible heading line
<b>c-c c-p</b>	Move to previous visible heading line
<b>c-c c-f</b>	Move to next visible heading line at same level
<b>c-c c-b</b>	Move to previous visible heading line at same level
<b>c-c c-u</b>	Move up to a lower-level visible heading line
<b>a-x hide-body</b>	Hide all body lines
<b>a-x show-all</b>	Show all lines
<b>c-c c-h</b>	Hide everything under this heading (excluding the heading)
<b>c-c c-s</b>	Show everything under this heading
<b>a-x hide-leaves</b>	Hide the body of this heading and all its subheadings.
<b>a-x show-leaves</b>	Show all subheadings of this heading.
<b>c-c c-i</b>	Show immediate subheadings
<b>a-x hide-entry</b>	Hide this heading's body.
<b>a-x show-entry</b>	Show this heading's body.

## Program Mode

A “sexp” is an expression, usually enclosed in some type of parenthesis, brackets, or braces. Lists are parenthetical groupings.

Key	Binding
<b>TAB</b>	Indent the “appropriate” amount for this language.
<b>c-a-f</b>	Move forward over a sexp
<b>c-a-b</b>	Move backward over a sexp
<b>c-a-k</b>	Kill sexp forward
<b>c-a-u</b>	Move up and backward in list structure
<b>c-a-d</b>	Move down and forward in list structure
<b>c-a-n</b>	Move forward over a list
<b>c-a-p</b>	Move backward over a list
<b>c-a-t</b>	Transpose expressions
<b>c-a-@</b>	Put mark after following expression
<b>c-a-a</b>	Move to beginning of current or preceding function
<b>c-a-e</b>	Move to end of current or following function
<b>c-a-h</b>	Put region around current or following function
<b>c-a-q</b>	Re-indent all lines within one list
<b>c-u TAB</b>	Shift list rigidly so that first line is properly indented
<b>c-a-\</b>	Re-indent all lines in the region
<b>a-;</b>	Insert or align comment
<b>c-x ;</b>	Set comment column
<b>c-u - c-x ;</b>	Kill comment on current line
<b>a-&lt;linefeed&gt;</b>	Return, followed by inserting and aligning comment
<b>a-(</b>	Put parenthesis around next sexp
<b>a-)</b>	Move past next close parenthesis and re-indent
<b>a-x add-change-log-entry</b>	Insert a change log entry (in ChangeLog file)

## C Indentation Variables

c-indent-level  
 c-continued-statement-offset  
 c-brace-offset  
 c-brace-imaginary-offset  
 c-argdecl-indent  
 c-label-offset

## Running Compilers under Emacs

Key	Binding
<b>a-x compile</b>	Run a compiler under Emacs, with messages to “*compilation*” buffer
<b>a-x grep</b>	Run grep under Emacs, list matching lines in “*compilation*” buffer
<b>a-x kill-compiler</b>	
<b>a-x kill-grep</b>	
<b>c-x `</b>	Visit the location of the next compiler message or grep match
<b>c-u c-x `</b>	Start at beginning of messages

## Compiler Variables

compile-command  
 Defaults to “make -l.” Can be overridden with local variables.

## Abbreviations

Key	Binding
<b>c-x +</b>	Define an abbrev to expand before point
<b>c-x c-a</b>	Define an abbrev that is only available in current major mode
<b>c-x -</b>	Define a word in buffer as an abbrev
<b>c-x c-h</b>	Define a word in buffer as a mode-specific abbrev
<b>a-x kill-all-abbrevs</b>	
<b>a-'</b>	Separate a prefix from an abbrev

<b>c-x ' </b>	Expand the abbrev before point
<b>a-x unexpand-abbrev</b>	Undo last abbrev expansion
<b>a-x expand-region-abbrevs</b>	Expand some or all abbrevs found in the region
<b>a-x list-abbrevs</b>	Print a list of all abbrevs
<b>a-x edit-abbrevs</b>	Edit a list of abbrevs
<b>a-x write-abbrev-file</b>	Write a file containing abbrevs
<b>a-x read-abbrev-file</b>	Read the abbrev file
<b>a-x define-abbrevs</b>	Define abbrevs from buffer
<b>a-x inster-abbrevs</b>	Insert all abbrevs and their expansions into the buffer

## Pictures

Key	Binding
<b>a-x edit-picture</b>	Enter picture-editing mode
<b>c-c c-c</b>	Return to previous major mode
<b>c-c &lt;</b>	Move left after insertion
<b>c-c &gt;</b>	Move right after insertion
<b>c-c ^</b>	Move up after insertion
<b>c-c .</b>	Move down after insertion
<b>c-c c-k</b>	Clear out the region. With argument, kill it.
<b>c-c c-w r</b>	Like c-c c-k, but save region in register r.
<b>c-c c-y</b>	Copy last killed rectangle into buffer by overwriting. Point defines upper left corner. With argument, insert instead of overwriting.
<b>c-c c-x r</b>	Like c-c c-y, but use rectangle from register r.

## Narrowing

Key	Binding
<b>c-x n</b>	Narrow down to the region
<b>c-x w</b>	Widen to entire buffer (make entire buffer visible)

## Sorting

Key	Binding
<b>a-x sort-lines</b>	Sort region. A prefix argument means sort into descending order.
<b>a-x sort-paragraphs</b>	Sort region by paragraphs.
<b>a-x sort-pages</b>	Sort region by pages.
<b>a-x sort-fields</b>	Sort region by lines. Fields are separated by whitespace. Field number is specified by argument. Negative argument signals descending order sort.
<b>a-x sort-numeric-fields</b>	Convert field to a number and sort.
<b>a-x sort-columns</b>	Sort region on fixed range of columns. Put “point” at one column and “mark” at the other column. Region of sort is defined as the lines “point” and “mark” are located in. This command sorts a rectangle.

## Shell Commands

Key	Binding
<b>a-!</b>	Run shell command
<b>a- </b>	Run shell command with region contents as input. Optionally replace region with output.
<b>a-x shell</b>	Run a subshell with input and output through Emacs buffer.

## SubShell Commands

Key	Binding
<b>RET</b>	Send buffer line as input (if at end of line), otherwise copy line to end of buffer and send it. Leave out prompt.
<b>c-c c-d</b>	Send end of file.
<b>c-c c-u</b>	Kill all text that hasn't been sent yet.

<b>c-c c-w</b>	Kill a word before point
<b>c-c c-c</b>	Interrupt the shell
<b>c-c c-z</b>	Stop the shell
<b>c-c c-\</b>	Send quit signal to shell
<b>c-c c-o</b>	Delete last batch of output from shell
<b>c-c c-r</b>	Scroll top of last batch of output to top of window
<b>c-c c-y</b>	Copy the previous bunch of shell input to before point.

## Shell Variables

shell-prompt-pattern

Should contain a regular expression that matches shell prompt. This allows Emacs to delete shell prompt before resubmitting a command.

## Printing Commands

Key	Binding
<b>a-x print-buffer</b>	Print copy of current buffer using print command
<b>a-x lpr-buffer</b>	Print copy of current buffer using lpr command
<b>a-x print-region</b>	Like print-buffer, but only prints region
<b>a-x lpr-region</b>	Like lpr-buffer, but only prints region

## Misc Commands

Key	Binding
<b>a-x top-level</b>	Exit recursive editing level
<b>a-x what-page</b>	Print page number and line in page
<b>a-x what-line</b>	Print line number
<b>a- =</b>	Number of lines in current region
<b>c-x =</b>	Print character code and position
<b>a-x doctor</b>	End each line by typing RET twice.
<b>a-x yow</b>	???

## **Ack! Ack!**

All of this material was extracted (painfully, I might add) from the *GNU Emacs Manual*, Fifth Edition, Emacs Version 18 for Unix Users (dated October 1986), and from the Epoch Release 3.1 documentation.

Kevin McCullen

