

Text Modes

Operation	Keystroke	Function	Note
Text Modes	Emacs has several other text modes that deal with whitespaces. See the Text Whitespace table for more information. Notes: <ul style="list-style-type: none">The superword-mode and subword-mode can be quite useful when editing source code where underscores are used to separate words in symbols or when CamelCase is being used.As of Emacs 23.2 the CC Mode c-subword-mode is obsolete and has been replaced by the more general subword-mode.		
Switch Insert/Overwrite mode	<ul style="list-style-type: none"><insert><f11> t o<f11> `	(overwrite-mode &optional ARG)	Toggles the overwrite mode on/off <ul style="list-style-type: none">The <insert> key is not available in macOS keyboards.With a prefix argument ARG, enable Overwrite mode if ARG is positive, and disable it otherwise.
Binary file overwrite only mode	<f11> t O	(nhexl-overwrite-only-mode &optional ARG)	Minor mode where text is only overwritten. Insertion/deletion is avoided where possible and replaced by overwriting existing text, if needed with 'nhexl-overwrite-clear-byte'. 👉 The nhexl mode must first be activated to edit the file in binary mode. 🔗📦 Requires the nhexl-mode package. 🔗🔗 PEL activates this when the pel-use-nhexl customize variable is set to t .
Make Info control text visible/invisible (toggle visible mode)	<f11> t m v	(visible-mode &optional ARG)	Toggle making all invisible text temporarily visible (Visible mode). With a prefix argument ARG, enable Visible mode if ARG is positive, and disable it otherwise. Useful for developing info files, where some characters are not visible by default. Same in Org-mode (for example to show everything, or to show the syntax of links without expanding anything).
Toggle subword-mode	<ul style="list-style-type: none"><f11> t m b<f12> M-b	(subword-mode &optional ARG)	Toggle subword-mode: a minor mode that treats sections of camelCase and PascalCase as distinct words. <ul style="list-style-type: none">With a prefix argument ARG, enable Subword mode if ARG is positive, and disable it otherwise.PEL provides the <f12> M-b key for the programming language modes where camelCase and PascalCase are popular.
Toggle superword-mode	<f11> t m p	(superword-mode &optional ARG)	Toggle superword-mode: a minor mode that treats snake_case as one word. In Lisp, '-' and '_' are treated part of words. <ul style="list-style-type: none">With a prefix argument ARG, enable Superword mode if ARG is positive, and disable it otherwise.
Toggle sentence separators between 1 or 2 spaces	<f11> t m s	(pel-toggle-sentence-end)	Toggle definition of end of sentence between 2 and 1 space character (to help text filling). This has an impact on the commands that deal with sentences (navigation such as (backward-sentence) and (forward-sentence), kill such as (kill-sentence) and (backward-kill-sentence).
Toggle local electric quote mode	<f11> t m ´	(electric-quote-local-mode &optional ARG)	Toggle ' electric-quote-mode ' only in this buffer. Useful to insert nicer-looking quote characters.
Show state of text modes	<f11> t m ?	(pel-show-text-modes)	Display the state of the various text modes in the mini buffer.
Text Whitespace Modes	The following Emacs command control how whitespace is shown or hidden. The following commands are also described in the Text Whitespace table.		
Toggle Whitespace Mode (Also in: ⌘ Whitespace)	<f11> t w m	(whitespace-mode &optional ARG)	Toggle whitespace visualization (Whitespace mode). With a prefix argument ARG, enable Whitespace mode if ARG is positive, and disable it otherwise. The kind of whitespace visualized is determined by the list variable <i>whitespace-style</i> , <i>whitespace-newline</i> .
Hide/Show trailing whitespaces (Also in: ⌘ Whitespace)	<f11> t w T	(pel-toggle-show-trailing-whitespace)	Toggle highlight of the trailing whitespaces in current buffer. Toggles the value of the variable <i>show-trailing-whitespace</i> .
Hide/Show trailing empty lines (Also in: ⌘ Whitespace)	<f11> t w e	(pel-toggle-indicate-empty-lines)	Toggle highlight of empty lines. Toggles the value of the variable <i>indicate-empty-lines</i> .
Toggle individual elements of whitespace-style (Also in: ⌘ Whitespace)	<f11> t w o	(whitespace-toggle-options ARG)	<ul style="list-style-type: none">If local whitespace-mode is off, toggle the option given by ARG and turn on local whitespace-mode.If local whitespace-mode is on, toggle the option given by ARG and restart local whitespace-mode. The argument, which is a single character and must be typed following the <f11> t w o , can be: <ul style="list-style-type: none">f toggle face visualizationt toggle TAB visualizations toggle SPACE and HARD SPACE visualizationr toggle trailing blanks visualizationl toggle "long lines" visualizationL toggle "long lines" tail visualizationn toggle NEWLINE visualizatione toggle empty line at bob and/or eob visualizationC-i toggle indentation SPACEs visualization (via 'indent-tabs-mode')I toggle indentation SPACEs visualizationi toggle indentation TABs visualizationC-t toggle big indentation visualizationC-a toggle SPACEs after TAB visualization (via 'indent-tabs-mode')A toggle SPACEs after TAB: SPACEs visualizationa toggle SPACEs after TAB: TABs visualizationC-b toggle SPACEs before TAB visualization (via 'indent-tabs-mode')B toggle SPACEs before TAB: SPACEs visualizationb toggle SPACEs before TAB: TABs visualization? Show the above list of options.
Mark and Region (Also in: ⌘ Marking)	With most editors when you type over a “selected” region, the text in the selection is automatically replaced by the new text. By default Emacs does not behave like this; instead it allows typing text while there is an active a marked region. If you want Emacs behave like other editors and automatically replace the text activate the “ <i>delete-selection-mode</i> ” with the following command.		
Toggles delete selection mode Also in: ⌘ Cut&Paste , Marking)	<f11> t m d	(delete-selection-mode)	Toggles delete selection-mode on/off. In delete-selection-mode typing a character while a region is active replaces the entire region with what is typed. By default delete selection-mode is off.
Drawing ASCII in Emacs	Emacs provides the picture-mode and artist-mode to draw ASCII-based pictures. Both are available when Emacs runs in graphics and terminal mode. However, I have not been able to use artist-mode with the mouse, even with xterm-mouse-mode active: each click just prints an ANSI sequence code.		
Picture Mode	Emacs supports the picture mode that allow you to move your cursor freely anywhere inside the window, which greatly simplify creating rectangular shapes for tables or even drawing ASCII-art. This work well in both graphics and terminal mode.		
Enter picture mode (See also: ⌘ Drawing)	<ul style="list-style-type: none"><f11> D p<f11> t p	(picture-mode)	Switch to Picture mode, in which a quarter-plane screen model is used. 👉 Very useful to type text in vertical fashion when for example, writing reStructuredText table. <ul style="list-style-type: none">Type C-c C-c to exit picture-mode and return to the mode previously used in the buffer.
Artist Mode	Although you can get some commands to work in terminal mode, it's best to use artist-mode when running Emacs in graphics mode.		
Toggle artist mode (See also: ⌘ Drawing)	<f11> D a	(artist-mode &optional ARG)	Toggle Artist mode. <ul style="list-style-type: none">With argument ARG, turn Artist mode on if ARG is positive.Artist lets you draw lines, squares, rectangles and poly-lines, ellipses and circles with your mouse and/or keyboard.

Text Modes — References

Topic & Link	Notes
GNU Emacs Manual: Text - Words	
GNU Emacs Manual: Text - Sentences	
GNU Emacs Manual: Text - Paragraphs	
GNU Emacs Manual: Text - Quotation Marks	
GNU Emacs Manual: Modes - Minor Modes	
GNU Emacs Manual: Programs - Other Features Useful for Editing Programs	
GNU Info Manual: Getting Started - Invisible text in Emacs Info	