

# Text Modes

Operation	Keystroke	Function	Note
<b>Text Modes</b>  (See also: <a href="#">☞ Whitespace</a> )	Emacs has several other text modes that deal with whitespaces. <ul style="list-style-type: none"><li>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.</li><li>As of Emacs 23.2 the CC Mode c-subword-mode is obsolete and has been replaced by the more general subword-mode.</li></ul>		
<b>Switch Insert/Overwrite mode</b>	<ul style="list-style-type: none"><li>&lt;f11&gt; t o</li><li>&lt;f11&gt; `</li></ul>	( <b>overwrite-mode</b> &optional ARG)	Toggles the overwrite mode on/off <ul style="list-style-type: none"><li>The &lt;insert&gt; key is not available in macOS keyboards.</li><li>With a prefix argument ARG, enable Overwrite mode if ARG is positive, and disable it otherwise.</li></ul>
<b>Binary file overwrite only mode</b>	<f11> t o	( <b>nhexl-overwrite-only-mode</b> &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 <a href="#">nhexl-mode</a> package. 🔗📧 PEL activates this when the <b>pel-use-nhexl</b> customize variable is set to <b>t</b> .
<b>Make Info control text visible/invisible (toggle visible mode)</b>	<f11> t m v	( <b>visible-mode</b> &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).
<b>Toggle subword-mode</b>  (See also: <a href="#">☞ Search/Replace</a> )	<ul style="list-style-type: none"><li>&lt;f11&gt; t m b</li><li>&lt;f12&gt; M-b</li><li>&lt;M-f12&gt; M-b</li></ul>	( <b>subword-mode</b> &optional ARG)	Toggle subword-mode: a minor mode that treats sections of <u>camelCase</u> and <u>PascalCase</u> as distinct words. <ul style="list-style-type: none"><li>With a prefix argument ARG, enable Subword mode if ARG is positive, and disable it otherwise.</li><li>PEL provides the &lt;f12&gt; M-b key for the programming language modes where <u>camelCase</u> and <u>PascalCase</u> are popular.</li></ul>
<b>Toggle superword-mode</b>  (See also: <a href="#">☞ Search/Replace</a> )	<ul style="list-style-type: none"><li>&lt;f11&gt; t m p</li><li>&lt;f12&gt; M-p</li><li>&lt;M-f12&gt; M-p</li></ul>	( <b>superword-mode</b> &optional ARG)	Toggle superword-mode: a minor mode that treats <u>snake_case</u> as one word. In Lisp, '-' and '_' are treated part of words. <ul style="list-style-type: none"><li>With a prefix argument ARG, enable Superword mode if ARG is positive, and disable it otherwise.</li><li>PEL provides the &lt;f12&gt; M-p key for the programming language modes where <u>snake_case</u> is popular (Emacs Lisp, C, C++, Erlang, Python, etc...)</li></ul>
<b>Toggle sentence separators between 1 or 2 spaces</b>	<f11> t m s	( <b>pel-toggle-sentence-end</b> )	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).
<b>Toggle local electric quote mode</b>	<f11> t m '	( <b>electric-quote-local-mode</b> &optional ARG)	Toggle 'electric-quote-mode' only in this buffer. Useful to insert nicer-looking quote characters.
<b>Show state of text modes</b>	<f11> t m ?	( <b>pel-show-text-modes</b> )	Display the state of the various text modes in the mini buffer.
<b>Text Whitespace Modes</b>	The following Emacs command control how whitespace is shown or hidden. The following commands are also described in the <a href="#">Text Whitespace table</a> .		
<b>Toggle Whitespace Mode</b>  (See also: <a href="#">☞ Whitespace</a> )	<f11> t w m	( <b>whitespace-mode</b> &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 <b>whitespace-style</b> , <b>whitespace-newline</b> .
<b>Hide/Show trailing whitespaces</b>  (See also: <a href="#">☞ Whitespace</a> )	<f11> t w T	( <b>pel-toggle-show-trailing-whitespace</b> )	Toggle highlight of the trailing whitespaces in current buffer.  Toggles the value of the variable <b>show-trailing-whitespace</b> .
<b>Hide/Show trailing empty lines</b>  (See also: <a href="#">☞ Whitespace</a> )	<f11> t w e	( <b>pel-toggle-indicate-empty-lines</b> )	Toggle highlight of empty lines.  Toggles the value of the variable <b>indicate-empty-lines</b> .
<b>Toggle individual elements of whitespace-style</b>  (See also: <a href="#">☞ Whitespace</a> )	<f11> t w o	( <b>whitespace-toggle-options</b> ARG)	<ul style="list-style-type: none"><li>If local whitespace-mode is off, toggle the option given by ARG and turn on local whitespace-mode.</li><li>If local whitespace-mode is on, toggle the option given by ARG and restart local whitespace-mode.</li></ul> 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"><li>f toggle face visualization</li><li>t toggle TAB visualization</li><li>s toggle SPACE and HARD SPACE visualization</li><li>r toggle trailing blanks visualization</li><li>l toggle "long lines" visualization</li><li>L toggle "long lines" tail visualization</li><li>n toggle NEWLINE visualization</li><li>e toggle empty line at bob and/or eob visualization</li><li>C-i toggle indentation SPACeS visualization (via 'indent-tabs-mode')</li><li>I toggle indentation SPACeS visualization</li><li>i toggle indentation TABs visualization</li><li>C-t toggle big indentation visualization</li><li>C-a toggle SPACeS after TAB visualization (via 'indent-tabs-mode')</li><li>A toggle SPACeS after TAB: SPACeS visualization</li><li>a toggle SPACeS after TAB: TABs visualization</li><li>C-b toggle SPACeS before TAB visualization (via 'indent-tabs-mode')</li><li>B toggle SPACeS before TAB: SPACeS visualization</li><li>b toggle SPACeS before TAB: TABs visualization</li><li>? Show the above list of options.</li></ul>
<b>Mark and Region</b>  ( See also: <a href="#">☞ Marking</a> )	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 “delete-selection-mode” with the following command.		
<b>Toggles delete selection mode</b>  (See also: <a href="#">☞ Cut&amp;Paste</a> , <a href="#">Marking</a> )	<f11> t m d	( <b>delete-selection-mode</b> )	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.
<b>Drawing ASCII in Emacs</b>	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.		
<b>Picture Mode</b>	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.		
<b>Enter picture mode</b>  (See also: <a href="#">☞ Drawing</a> )	<ul style="list-style-type: none"><li>&lt;f11&gt; D p</li><li>&lt;f11&gt; t p</li></ul>	( <b>picture-mode</b> )	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"><li>Type C-c C-c to exit picture-mode and return to the mode previously used in the buffer.</li></ul>
<b>Artist Mode</b>	Although you can get some commands to work in terminal mode, it's best to use artist-mode when running Emacs in graphics mode.		
<b>Toggle artist mode</b>  (See also: <a href="#">☞ Drawing</a> )	<f11> D a	( <b>artist-mode</b> &optional ARG)	Toggle Artist mode. <ul style="list-style-type: none"><li>With argument ARG, turn Artist mode on if ARG is positive.</li><li>Artist lets you draw lines, squares, rectangles and poly-lines, ellipses and circles with your mouse and/or keyboard.</li></ul>

Text Modes — References

Topic & Link	Notes
<a href="#">GNU Emacs Manual: Text - Words</a>	
<a href="#">GNU Emacs Manual: Text - Sentences</a>	
<a href="#">GNU Emacs Manual: Text - Paragraphs</a>	
<a href="#">GNU Emacs Manual: Text - Quotation Marks</a>	
<a href="#">GNU Emacs Manual: Modes - Minor Modes</a>	
<a href="#">GNU Emacs Manual: Programs - Other Features Useful for Editing Programs</a>	
<a href="#">GNU Info Manual: Getting Started - Invisible text in Emacs Info</a>	