

# Transpose Text

Operation	Keystroke	Function	Note
<a href="#">Transpose (swap) text elements</a>	Emacs supports the following commands that swap a text element with the preceding one. <ul style="list-style-type: none"> <li>• PEL provides extra key bindings, available for convenience and discovery via which-key. The Emacs native keys are shorter.</li> <li>• PEL also provides key bindings for more native Emacs commands.</li> </ul>		
Last updated on:	2025-09-16		
Open this PDF file. See also: <a href="#">🔗 Help/Info</a>	<a href="#">&lt;f11&gt; t t &lt;f1&gt;</a>	<a href="#">(pel-help-pdf &amp;optional OPEN-WEB-PAGE)</a>	Open the <a href="#">🔗 Transpose</a> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around.
<a href="#">Transpose two characters</a>	<ul style="list-style-type: none"> <li>• <b>C-t</b></li> <li>• <a href="#">&lt;f11&gt; t t c</a></li> </ul>	<b>(transpose-chars ARG)</b>	Interchange characters around point, moving forward one character. <ul style="list-style-type: none"> <li>• With prefix arg ARG, effect is to take character before point and drag it forward past ARG other characters (backward if ARG negative).</li> <li>• If no argument and at end of line, the previous two chars are exchanged.</li> </ul>
<a href="#">Transpose two words</a>	<ul style="list-style-type: none"> <li>• <b>M-t</b></li> <li>• <a href="#">&lt;f11&gt; t t w</a></li> </ul>	<b>(transpose-words ARG)</b>	Interchange words around point, leaving point at end of them. <ul style="list-style-type: none"> <li>• With prefix arg ARG, effect is to take word before or around point and drag it forward past ARG other words (backward if ARG negative).</li> <li>• If ARG is zero, the words around or after point and around or after mark are interchanged.</li> </ul> <p>To use: put cursor between 2 words to transpose.</p> <ul style="list-style-type: none"> <li>• After transposition emacs leaves the cursor after the second word.</li> <li>• Emacs transpose words, but not punctuation. The concept of word is, however, flexible because it is controlled by Emacs syntax tables.</li> </ul> <p>👉 See <a href="#">Emacs Text - Words</a> section for more information on how to operate on words with Emacs. Also see <a href="#">🔗 Text Modes</a> .</p>
<b>Transpose two sub-words</b> <ul style="list-style-type: none"> <li>• <b>subword-mode</b> must be on</li> <li>• 2 words can be separated by dash, underscore or the 2 parts of a camelCase word.</li> <li>• See: <a href="#">🔗 Text Modes</a></li> </ul>	<a href="#">&lt;f11&gt; t t S</a>	<b>(subword-transpose ARG)</b>	Do the same as <b>'transpose-words'</b> but on subwords. <ul style="list-style-type: none"> <li>• See the command <b>'subword-mode'</b> in <a href="#">🔗 Text Modes</a> for a description of subwords.</li> <li>• Optional argument ARG is the same as for <b>'transpose-words'</b>.</li> </ul> <p>👉 Use this transpose words when the subword-mode is active. Useful to transpose words separated by dash, underscores or camelCase words.</p>
<a href="#">Transpose two lines</a>	<ul style="list-style-type: none"> <li>• <b>C-x C-t</b></li> <li>• <a href="#">&lt;f11&gt; t t l</a></li> </ul>	<b>(transpose-lines ARG)</b>	Exchange current line and previous line, leaving point after both. <ul style="list-style-type: none"> <li>• With argument ARG, takes previous line and moves it past ARG lines.</li> <li>• With argument 0, interchanges line point is in with line mark is in.</li> </ul> <p>To use: put cursor to the second line.</p>
<a href="#">Transpose two balanced expressions (sexps)</a>	<ul style="list-style-type: none"> <li>• <b>C-M-t</b></li> <li>• <a href="#">&lt;f11&gt; t t x</a></li> </ul>	<b>(transpose-sexps ARG)</b>	Transpose 2 balanced expressions (text enclosed in parenthesis, braces, square or angle brackets, quotes, back-quotes and double quotes) of the same of different types. Here they are globally identified as <i>sexpr</i> . <ul style="list-style-type: none"> <li>• Unlike 'transpose-words', point must be between the two sexps and not in the middle of a sexp to be transposed.</li> <li>• With non-zero prefix arg ARG, effect is to take the sexp before point and drag it forward past ARG other sexps (backward if ARG is negative).</li> <li>• If ARG is zero, the sexps ending at or after point and at or after mark are interchanged.</li> </ul>
<b>Transpose two sentences</b>	<a href="#">&lt;f11&gt; t t s</a>	<b>(transpose-sentences ARG)</b>	Interchange the current sentence with the next one. <ul style="list-style-type: none"> <li>• With prefix argument ARG a non-zero integer, moves the current sentence past ARG sentences, leaving point after the current sentence.</li> <li>• If ARG is positive, moves the current sentence forwards, if ARG is negative moves it backwards. If ARG is zero, exchanges the current sentence with the one containing the mark.</li> </ul> <p>Note that by default sentences must be separated by end of line or 2 space characters. The number of space characters required can be modified. The PEL package provides the (pel-toggle-sentence-end) command, bound to <a href="#">&lt;f11&gt; t m s</a> to toggle the value from 2 to 1. See the <b>Text Modes</b> table.</p> <p>👉 See <a href="#">Emacs Text - Sentences</a> section for more information on how Emacs defines a sentence and the commands dealing with them.</p>
<b>Transpose two paragraphs</b>	<a href="#">&lt;f11&gt; t t p</a>	<b>(transpose-paragraphs ARG)</b>	Interchange the current paragraph with the next one. <ul style="list-style-type: none"> <li>• With prefix argument ARG a non-zero integer, moves the current paragraph past ARG paragraphs, leaving point after the current paragraph.</li> <li>• If ARG is positive, moves the current paragraph forwards, if ARG is negative moves it backwards. If ARG is zero, exchanges the current paragraph with the one containing the mark.</li> </ul> <p>👉 See <a href="#">Emacs Text - Paragraphs</a> section for more information on how Emacs define a paragraph and the commands dealing with them.</p>
<b>Transpose 2 regions</b>	<a href="#">&lt;f11&gt; t t r</a>	<b>(transpose-regions STARTR1 ENDR1 STARTR2 ENDR2 &amp;optional LEAVE-MARKERS)</b>	Transpose region STARTR1 to ENDR1 with STARTR2 to ENDR2. <ul style="list-style-type: none"> <li>• The regions should not be overlapping, because the size of the buffer is never changed in a transposition.</li> <li>• Optional fifth arg LEAVE-MARKERS, if non-nil, means don't update any markers that happen to be located in the regions.</li> <li>• Transposing beyond buffer boundaries is an error.</li> <li>• Interactively, STARTR1 and ENDR1 are point and mark; STARTR2 and ENDR2 are the last two marks pushed to the mark ring; LEAVE-MARKERS is nil.</li> <li>• If a prefix argument N is given, STARTR2 and ENDR2 are the two successive marks N entries back in the mark ring.</li> <li>• A negative prefix argument instead counts forward from the oldest mark in the mark ring.</li> </ul> <p>To use interactively:</p> <ol style="list-style-type: none"> <li>1. First move point at the beginning of the first region.</li> <li>2. Type <b>C-SPC</b> and move point to the end of the first region.</li> <li>3. Type <b>C-g</b></li> <li>4. Move point to the beginning of the second region.</li> <li>5. Type <b>C-SPC</b> and move point to the end of the second region.</li> <li>6. Execute the command.</li> </ol>

Transpose — References

Topic & Link	Notes
<a href="#">GNU Emacs Manual: Fixit - Transposing Text</a>	
<a href="#">GNU Emacs Manual: Text - Words</a>	
<a href="#">GNU Emacs Manual: Text - Sentences</a>	
<a href="#">GNU Emacs Manual: Text - Paragraphs</a>	