Case Conversions

Operation	Keystroke	Function	Note
Upper/Lower/ Capital(Title) Case See also: ∑ **Text Modes**			Emacs has 3 types of case conversion commands: to **u**pper case, to **l**ower case and to **c**apital case (also called title case) where the first letter of a word is upper case and all the other letters are lower case. • The main commands operate on words a concept that Emacs control via its syntax tables. What constitutes a word may be modified by changing the text mode, using the superword-mode or the subword-mode. The commands also operate on the letter where point (the cursor) is located. You need to position point at the beginning of the word to apply the conversion to all letters of the word. • You can apply the word-related commands to several words, before and after point, using command prefixes. • Emacs also provide commands that operate on a marked region. ☛ The PEL package provides more functionality with less key bindings. Plain Emacs uses **M-l**, **C-x C-l**, **M-u**, **C-x C-u**, and **M-c** for the case conversion operations. The PEL package uses only **M-c**, **M-c** and **M-T** key bindings which operate on the current word like the original commands, but also operate on the region if it is visible. They also operate on the complete word at point, regardless of point's position within the word if a 0 numerical argument is specified. The **M-c** and **M-C** commands capitalize word(s) when you use them to convert to the same case of the "criteria" character(s). This way if text is in lower case, you can capitalize each word using **M-c** and if text is in uppercase you can capitalize the word(s) with **M-C**. You can perform all conversions with **M-c** and **M-C,** no other key is really required. ☛ Note that **M-T** and **M-C** require the use of the Shift key. It's OK since the original commands break the region marking done with Shift pressed during navigation anyway. PEL frees the unused **M-l**, **M-u** for other bindings. The **C-x C-l** and **C-x C-u** provide the same but extended functionality.
Open this PDF file. See also: ∑ **Help/Info**	**<f11> t <f1> l**	**(pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the ∑ **Case Conversions** local PDF. If the prefix argument (like **C-u** or **M--**) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
Lowercase word(s) or all words in marked region ★ PEL Enhanced Key ★	• **M-c** • **<f11> t l**	**(pel-downcase-word-or-region** &optional N)	Downcase word(s) or all words in marked region. ⌨ Uses: *downcase-word* ☞ ✂ This differs from the original Emacs binding for **M-c**. • If no region marked, with: • no argument: downcase current word (from current position). • N = 0: downcase current word (from start of current word) • N > 0: downcase current word, N-1 following words and move forward. • N < 0: downcase current word and N-1 previous words. For non-zero N, the conversion starts at point, not at the beginning of the word (unless point is at beginning of word). • If region is marked: downcase all words in region. • Note: If "criteria" character is already a lower-case letter, the function capitalizes the word(s) instead. The "criteria" character is the first (or last) letter character at point of after (before) point depending of N's sign. This allows to use the key binding to downcase or capitalize a word. ☝ Use the prefix numerical argument to control the text conversion. For example: • **M-2 M-c** Convert current word and next word to lower case (or capitalized if the first word is already lower-case) • **M-- M-c** Convert last word to lower case (or to capitalized if that word is already lower case). Use this for fixing the case of the last typed word (but the keys differ from standard Emacs.) • **M-0 M-c** Converts the complete current word to lower case (or to capitalized if it already is lower-case), regardless of point position within the word. • **M-c** Converts the text starting at point to the end of current word to lower case (or capitalized if it already is lower-case).
Uppercase word(s) or all words in marked region ★ PEL Enhanced Key ★	• **M-C** • **<f11> t u**	**(pel-upcase-word-or-region** &optional N)	Upcase word(s) or all words in marked region. ⌨ Uses: *upcase-word* • If region is marked: upcase all words in region. • If no region marked, with: • no argument: upcase current word (from current position). • N = 0: upcase current word (from start of current word) • N > 0: upcase current word, N-1 following words and move forward. • N < 0: upcase current word and N-1 previous words. For non-zero N, the conversion starts at point, not at the beginning of the word (unless point is at beginning of word). • Note: If 2 "criteria" characters are already upper-case letters, the function capitalizes the word(s) instead. The two "criteria" characters are the first (or last) two letter characters at point of after (before) point depending of N's sign. This allows to use the key binding to downcase or capitalize a word. ☞Press the **Shift** key (with **M-c**) to execute this command. ☝ Use the prefix numerical argument to control the text conversion. For example: • **M-2 M-C** Convert current word and next word to upper case (or capitalized if the first word is already upper-case) • **M-- M-C** Convert last word to upper case (or to capitalized if that word is already upper case). Use this for fixing the case of the last typed word (but the keys differ from standard Emacs.) • **M-0 M-C** Converts the complete current word to upper case (or to capitalized if it's already upper-case), regardless of point position within the word. • **M-C** Converts the text starting at point to the end of current word to upper case (or capitalized if it's already upper-case).
Title-case word(s) or all words in marked region ★ PEL Enhanced Key ★ ⚠️ The **M-T** binding might be reassigned for other functionality in the future since the operation can be achieved via **M-c** or **M-C**.	• **M-T** • **<f11> t c**	**(pel-capitalize-word-or-region** &optional N)	Capitalize N word(s) or all words in marked region. ☞ Press the **Shift** key (with **M-t**) to execute this command. • If no region marked, with: • no argument: capitalize current word (from current position). • N = 0: capitalize current word (from start of current word) • N > 0: capitalize current word, N-1 following words and move forward. • N < 0: capitalize current word and N-1 previous words. For non-zero N, the conversion starts at point, not at the beginning of word (unless point is at beginning of word). Uses: *capitalize-word*. • If region is marked: • if no prefix (N is nil): convert first letter of each word to upper/title case and convert all other word letters to lower case, using `capitalize-region'. • if N <0: convert first letter of each word to upper/title case but leave the rest unchanged. Uses *upcase-initials-region*.
Letter(s) case change			Use the following two key bindings to change the case of a single letter at point. The commands can also be used with numerical arguments to specify the number of consecutive letters at point. They can also be used to change the case of the marked region just as the functions normally bound by vanilla Emacs do.
Upcase-region or single letter(s) at point ★ PEL Enhanced Key ★	**C-x C-u**	**(pel-upcase-letter** &optional N)	Convert letter(s) or region to upper case. • If a region is active, convert all letters in that region to upper case. • If no region is active converts N letters at point to uppercase. N defaults to 1. Specify a larger number with numerical argument to convert more than 1 letter. Negative numbers convert letter before point.
Lowercase-region or single letter(s) at point ★ PEL Enhanced Key ★	**C-x C-l**	**(pel-downcase-letter** &optional N)	Convert letter(s) or region to lower case. • If a region is active, convert all letters in that region to lower case. • If no region is active converts N letters at point to lowercase. N defaults to 1. Specify a larger number with numerical argument to convert more than 1 letter. Negative numbers convert letter before point.
Modified key bindings			⚠️ The following bindings exists in basic Emacs, but PEL re-assigns them and these key bindings are used for other operations. At the moment PEL rebinds the ones identified in red, the others are still available.
Uppercase word ✂ **M-u** is undo	**M-u**	**(upcase-word** ARG)	Convert to upper case from point to end of word, moving over. • If point is in the middle of a word, the part of that word before point is ignored when moving forward. • With negative argument, convert previous words but do not move.
Upcase-region ✂ **C-x C-u** is pel-upcase-letter	**C-x C-u**	**(upcase-region** BEG END &optional REGION-NONCONTIGUOUS-P)	Convert the region to upper case. • With PEL you can also do this with M-c. • Note that **pel-upcase-letter** behaves like **upcase-region** when a region is active, but it also works on N specified (default to 1) single letters.
Lowercase word ✂ **M-l** is auto-complete	**M-l**	**(downcase-word** ARG)	Convert to lower case from point to end of word, moving over. • If point is in the middle of a word, the part of that word before point is ignored when moving forward. • With negative argument, convert previous words but do not move.

Operation	Keystroke	Function	Note
Lowercase-region ✂ `C-x C-l` is **pel-downcase-letter**	`C-x C-l`	(**downcase-region** BEG END &optional REGION-NONCONTIGUOUS-P)	Convert the region to lower case. • With PEL you can also do this with M-C. • Note that **pel-downcase-letter** behaves like **downcase-region** when a region is active, but it also works on N specified (default to 1) single letters.
Capitalize word ✂ **PEL rebinds this key, see above.**	`M-c`	(**capitalize-word** ARG)	Capitalize from point to the end of word, moving over. • With numerical argument ARG, capitalize the next ARG-1 words as well. • This gives the word(s) a first character in upper case and the rest lower case. • If point is in the middle of a word, the part of that word before point is ignored when moving forward. • With negative argument, capitalize previous words but do not move.