Insert Control/Unicode Characters - Input Method - File encoding

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> |
|---|---|--|--|
| Input Methods | | language other than English, you ernate character input method to | |
| | _ | | he input natural language and the way to type non-ASCII characters. |
| Open this PDF file. See also: <u>N Help/Info</u> | <f11> t <f1> 2</f1></f11> | (pel-help-pdf &optional OPEN-WEB-PAGE) | Open the <u>Normal Method</u> 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. |
| • Changing Input-Method See also: <u>Ney-Chords</u> | Use the following commands to change the input-method when you want to type non-English characters such as accentuated letters used in several Latin and Germanic languages and several others. Changing input-method may prevent the key-chord mechanism to work properly. If it happens, toggle the key-chord-mode off and back on. | | |
| Enable/Disable selected input method for all buffers | <f11> t i</f11> | (toggle-input-method &optional ARG INTERACTIVE) | First time used, if no alternate input method has been selected, then prompts for another input method. For example, this can be used to select a French input method for example, one of french-prefix or french-postfix • Once an alternate input method is in effect, issuing this command again, restores the default input method. • With C-u prefix, always prompt. |
| Select a new input method for the current buffer | • C-x RET C-\ • <f11> t I</f11> | (set-input-method INPUT- METHOD &optional INTERACTIVE) | Set the input method for the current buffer only. • Always prompts. |
| Get Help on Input Method Substitute Subs | • C-h I • C-h C-\ • <f1> I</f1> | (describe-input-method INPUT-METHOD) | Prompts for the input method and then opens the help describing how the input method works. For instance if we identify the input method as french-postfix the help describes how to enter the accentuated characters with this input method. |
| Display a list of all supported input methods | <f11> ? d i</f11> | (list-input-methods) | Lists all input methods supported. Like french-prefix, TeX, IPA, etc |
| Describe encoding system Describe buffers encoding ➡ See also: <u>N Help/Info</u> | • C-h C • <f1> C • <f11> ? d C</f11></f1> | (describe-coding-system CODING-SYSTEM) | Display information about CODING-SYSTEM. • Prompts for coding system name. Supports completion. • Type RET to describe current buffer encoding. |
| Query info about point Show information about current character. See also: Help/Info | • C-x = • <f11> ? d p</f11> | (what-cursor-position &optional DETAIL) | Displays information about character at point in the echo area. With any prefix argument opens a *Help* buffer and show the complete information of character at point with all properties, face, encoding, etc. Type: C-u C-x = With PEL, you can also type: C C-x = |
| Display ASCII table See also: <u>Nelp/Info</u> | <f11> ? A</f11> | (ascii-table) | Show an interactive ASCII table in the other (next) window. Requires the <u>ascii-table</u> package PEL activates this when the <u>pel-use-ascii-table</u> user option is set to t. |
| Insert quoted character | C-q | (quoted-insert ARG) | Read next input character and insert it. This is useful for inserting control characters like hard-tab. Type number digit keys than something else to enter character by value. By default the value is entered in octal. But this can be changed by setting the read-quoted-char-radix variable to 10 for decimal or 16 for hexadecimal. With argument, insert ARG copies of the character. |
| Insert Special Quote characters | When writing non-ASCII text, use the following to insert curved quote instead of the default ASCII single and double quote characters. • Emacs supports the electric-quote-mode to convert the ASCII quote characters to curved quotes. • You can use the <f11> t m ' key binding to toggle the local electric-quote-mode (see the Text Mode table). • Without activating the electric-quote-mode you can also use the following commands to insert these curved quote characters.</f11> | | |
| Insert Curved Single Opening Quote | C-x 8 [| (self-insert-command N) | Inserts ' |
| Insert Curved Single Closing Quote | C-x 8] | (self-insert-command N) | Inserts ' |
| Insert Curved Double Opening Quote | C-x 8 { | (self-insert-command N) | Inserts " |
| Insert Curved Double Ending Quote | C-x 8 } | (self-insert-command N) | Inserts " |
| Insert accented letters | | | ters (although specialized input methods can also be used, like French input method or ve). The following are just examples showing the accent selectors. |
| Insert accent: grave | С-х 8 ` е | (self-insert-command N) | Inserts: è |
| Insert accent: acute | С-х 8 ′ е | (self-insert-command N) | Inserts: é |
| Insert accent: circumflex | С-ж 8 ^ е | (self-insert-command N) | Inserts: ê |
| Insert accent: tilde | C-x 8 - n | (self-insert-command N) | Inserts: ñ |
| Insert accent: umlaut | C-x 8 " u | (self-insert-command N) | Inserts: ü |
| Insert accent: cedilla | C-x 8 , c | (self-insert-command N) | Inserts: ç |
| Insert Chars by Name/value Insert arbitrary character by name | C-x 8 RET | (insert-char CHARACTER | Insert COUNT copies of CHARACTER. |
| Insert <u>Unicode character by name</u> → | | &optional COUNT INHERIT) ER. You can specify CHARACTE aracter name e g "I ATIN SMA | |
| Insert Unicode character by value ➡ | as its <u>Unicode character name</u>, e.g. "LATIN SMALL LETTER A". Completion is available; if you type a substring of the name preceded by an asterisk '*', Emacs shows all names which include that substring, not necessarily at the beginning of the name, as a hexadecimal code point, e.g. 263A. Note that code points in Emacs are equivalent to Unicode up to 10FFFF (which is the limit of the Unicode code space), | | |
| ■ In macOS you may also want to: | | | g. #o21430 (octal), #x2318 (hex), or #10r8984 (decimal). |
| add Greek as a Keyboard Input Sources and temporary select it to enter Greek characters directly. | Example: to insert the | Lambda greek letter λ you can ty 03bb <ret> GREEK SMALL LETTER LAMBI</ret> | • |
| Insert Greek Letters: using TeX input method in any OS | | r other mathematical symbols, u | use the TeX input method (<f11> t I TeX or C-u <f11> t i TeX).</f11></f11> |
| List all TeX char sequences with <f1> I TeX</f1> | \Alpha A \Beta B \Delta Δ | ming revolutes produce dreek | ^\beta β \Stigma ζ \sum Σ \forall ∀ |
| | \lambda λ \gamma γ | | \frac25 \frac25 |

| <u>Operation</u> | <u>Keystroke</u> | Function | Note |
|---|--------------------------------|---|--|
| Insert Greek Letter | <f6> g 🕸</f6> | Insert a greek letter by typing equivalent Greek letter. | <f6> g followed by a key in [a-zA-Z] range inserts the Unicode character for the</f6> |
| • With the <f6> g prefix</f6> | | This PEL key binding is alw Examples: | ays available. |
| See also: <u>∑ Inserting Text</u> | | <f6> g a inserts a</f6> | <f6> g b inserts β <f6> g A inserts A</f6></f6> |
| | | <f6> g B inserts B The insertions work everyw To see all keys:</f6> | <f6> g 1 inserts A where insert is allowed, including in response to prompts.</f6> |
| | | • Type <f6> g C-h</f6> | ad in the second |
| Insert Greek Letter | <f9> (*)</f9> | Use <u>which-key mode</u> ar Insert a greek letter by typing | <f9> followed by a key in [a-zA-Z] range inserts the Unicode character for the</f9> |
| With <f9> key: Available only when the</f9> | | equivalent Greek letter. Only available when pel-actions are selected in the | ctivate-f9-for-greek user-option is turned on. |
| pel-activate-f9-for-greek user- option is turned on. | | Examples: <f9> a inserts a <f< td=""><td>9> b inserts β <f9> A inserts A <f9> B inserts B <f9> 1 inserts λ</f9></f9></f9></td></f<></f9> | 9> b inserts β <f9> A inserts A <f9> B inserts B <f9> 1 inserts λ</f9></f9></f9> |
| See also: Namerting Text | | The insertions work everywTo see all keys: | here insert is allowed, including in response to prompts. |
| | | Type <f9> C-h</f9>Use which-key mode ar | nd just type <f9></f9> |
| | | | ound to a key: it's an additional set of bindings added to Emacs key-translation-map. |
| • Quickly type succession of Greek | <f7> <f6> <f6></f6></f6></f7> | | <f6> type one of the Meta letter keys in the hydra to insert a Greek character. Type</f6> |
| characters | | character, etc | rt them, latin letters, digits, punctuation characters, and Meta-char to inter the greek r keys may not work because they are often encoded using Esc keys with is mapped to |
| See also: <u>∑ Inserting Text</u> | | Meta. | |
| | | | nal package. dativated by pel-use-hydra. ivate-hydra-for-greek to t to activate this hydra. |
| | | Exit the hydra by typing <f7< li=""> • • • </f7<> | 7> |
| | | | pols Defs Lisp-Interaction Help |
| | | | oe Greek text by pressing the Meta keys. να πληκτρολογήσετε ελληνικό κείμενο πατώντας τα |
| | | | git keys can be typed: 0123456789 lable: use `` <fil> u u`` instead.</fil> |
| | | | sor keys may not work though. Use C-b, C-f, C-n, C-p instead. |
| | | | All (9,29) (Lisp Interaction ω th WK Anzu Fly ² ElDoc) 12:56pm, χ, [M-d]: δ, [M-e]: ε, [M-f]: φ, [M-g]: γ, [M-h]: η, [M-i]: ι, [M-j]: φ, [M-k]: κ, |
| | | [M-1]: λ, [M-m]: μ, [M-n]: ν [M-x]: ξ, [M-y]: ψ, [M-z]: δ | ν, [M-o]: ο, [M-p]: π, [M-q]: θ, [M-x]: ρ, [M-s]: σ, [M-t]: τ, [M-u]: υ, [M-w]: ω, ΄ ζ, [M-A]: A, [M-B]: Β, [M-C]: Χ, [M-D]: Δ, [M-E]: Ε, [M-F M-J]: Φ, [M-G]: Γ, [M-H]: ΄ |
| | | |]: Λ, [M-M]: M, [M-N]: N, [M-O]: O, [M-P]: Π, [M-Q]: Θ, [M-R]: P, [M-S]: Σ, [M-T]: {]: Ξ, [M-Y]: Ψ, [M-Z]: Z, [<f7>]: cancel.</f7> |
| Some of the special characters that can be inserted with C-x 8 keys | C-x 8 ! C-x 8 \$ | i ¤ | C-x 8 ~ A |
| our se moercea mar e x e nege | C-x 8 + C-x 8 - | ± | C-x 8 - N Ñ C-x 8 - O Õ |
| | C-x 8 . C-x 8 < | • | C-x 8 ~ T |
| | C-x 8 = C-x 8 > | - » | C-x 8 ~ n ñ C-x 8 ~ o ō |
| | C-x 8 ? | ¿ © | C-x 8 ~ t |
| | C-x 8 L C-x 8 P | £ | C-x 8 ` A À C-x 8 ` E È |
| | C-x 8 R | ® § | C-x 8 ` O |
| | C-x 8 Y | ¥ | C-x 8 ` a à à C-x 8 ` e è |
| | С-х 8 о | 0 | C-x 8 ` i |
| Some of the special characters that can be inserted with C-x 8 keys | C-x 8 u C-x 8 x | μ × | C-x 8 ` o |
| (continued). | C-x 8 a < C-x 8 a = | ← ↔ | C-x 8 _ < ≤ C-x 8 _ > ≥ |
| | C-x 8 a > C-x 8 N o | → | C-x 8 _ a |
| | C-x 8 2 + C-x 8 - SPC | ‡ ~ | C-x 8 ^ a â C-x 8 ^ e ê |
| | C-x 8 ~ = C-x 8 / / | ≈ ÷ | C-x 8 " a ä C-x 8 " e ë |
| | C-x 8 / = C-x 8 3 / 4 | ≠ 3 ₄ | C-x 8 " i |
| | C-x 8 1 / 2 | 1/2 | C-x 8 " s ß C-x 8 " u ü |
| | C-x 8 1 / 4 C-x 8 ^ 1 | 1 ₄ 1 | C-x 8 , C Ç |
| | C-x 8 ^ 2 C-x 8 ^ 3 | 3 | C-x 8 , c |
| File Encoding | C-x 8 * * The following comman | ds and functions can help investigations | C-x 8 / o ø stigate and handle file encoding issues. |
| File Encoding Show encoding of file visited in | <f11> ? d e</f11> | (pel-show-buffer-file- | Show coding system of file in current buffer. |
| • See also: <u>Nelp/Info</u> | | encoding) | Open a *Help* buffer and show the value of the buffer-file-coding-system variable. |
| Query info about point | • C-x = • <f11> ? d p</f11> | (what-cursor-position &optional DETAIL) | Displays information about character at point in the echo area: position, character, encoding. |
| • See also: <u>N Help/Info</u> | | | With any prefix argument opens a *Help* buffer and show the complete information of character at point with all properties, face, encoding, etc. |
| | | | Type: C-u C-x = With PEL, you can also type: C C-x = |
| | | 1 | 77 ener 26 |

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> |
|---|--------------------------------------|--|--|
| Visit a file literally: with no encoding support and conversion See also: File-mngt | <f11> f M-1</f11> | (find-file-literally FILENAME) | Visit file FILENAME with no conversion of any kind. Format conversion and character code conversion are both disabled, and multibyte characters are disabled in the resulting buffer. The major mode used is Fundamental mode regardless of the file name, and local variable specifications in the file are ignored. Automatic uncompression and adding a newline at the end of the file due to 'require-final-newline' is also disabled. If Emacs already has a buffer which is visiting the file, this command asks you whether to visit it literally instead. |
| <u>Change text encoding</u> of a region or the entire file. | M-x recode- region | (recode-region START END NEW-CODING CODING) | Re-decode the region (previously decoded by CODING) by NEW-CODING. Prompts for the new and current encoding. Supports tab completion when selecting an encoding. |
| Set coding system for coding/ decoding file names | C-x RET F | (set-file-name-coding- system CODING-SYSTEM) | Set coding system for decoding and encoding file names to CODING-SYSTEM. • It actually just set the variable 'file-name-coding-system' to CODING-SYSTEM. |
| Select coding system for next communication with other window system clients. | C-x RET X | (set-next-selection-coding- system CODING-SYSTEM) | Use CODING-SYSTEM for next communication with other window system clients. • This setting is effective for the next communication only. |
| Execute an I/O command with specified coding system | C-x RET C | (universal-coding-system- argument CODING-SYSTEM) | Execute an I/O command using the specified coding system. Prompts for the encoding (supports tab completion), then wait for the Emacs command to execute. |
| Set the coding system of the file visited by current buffer - take effect on next file save. Solution To change line-endings of a file, use on of the following: • C-x RET f unix → unix -style • C-x RET f dos → dos-style • C-x RET f mac → mac-style | C-x RET f | (set-buffer-file-coding- system CODING-SYSTEM &optional FORCE NOMODIFY) | Set the file coding-system of the current buffer to CODING-SYSTEM. This means that when you save the buffer, it will be converted according to CODING-SYSTEM. For a list of possible values of CODING-SYSTEM, use M-x list-coding-systems. If CODING-SYSTEM leaves the text conversion unspecified, or if it leaves the end-of-line conversion unspecified, FORCE controls what to do. If FORCE is nil, get the unspecified aspect (or aspects) from the buffer's previous 'buffer-file-coding-system' value (if it is specified there). Otherwise, leave it unspecified. This marks the buffer modified so that the succeeding C-x C-s surely saves the buffer with CODING-SYSTEM. From a program, if you don't want to mark the buffer modified, specify t for NOMODIFY. If you know exactly what coding system you want to use, just set the variable 'buffer-file-coding-system' directly. |
| Set coding system for terminal keyboard | C-x RET k | (set-keyboard-coding- system CODING-SYSTEM &optional TERMINAL) | Set coding system for keyboard input on TERMINAL to CODING-SYSTEM. For a list of possible values of CODING-SYSTEM, use M-x list-coding-systems. The default is determined by the selected language environment or by the previous use of this command. If CODING-SYSTEM is nil or the coding-type of CODING-SYSTEM is 'raw-text', the decoding of keyboard input is disabled. TERMINAL may be a terminal object, a frame, or nil for the selected frame's terminal. The setting has no effect on graphical terminals. |
| Set tup multilingual environment | C-x RET 1 | (set-language-environment LANGUAGE-NAME) | Set up multilingual environment for using LANGUAGE-NAME. • This sets the coding system priority and the default input method and sometimes other things. LANGUAGE-NAME should be a string which is the name of a language environment. For example, "Latin-1" specifies the character set for the major languages of Western Europe. If there is a prior value for 'current-language-environment', this runs the hook 'exit-language-environment-hook'. After setting up the new language environment, it runs 'set-language-environment-hook'. |
| Set coding for process associated with current buffer | C-x RET p | (set-buffer-process-coding- system DECODING ENCODING) | Set coding systems for the process associated with the current buffer. DECODING is the coding system to be used to decode input from the process, ENCODING is the coding system to be used to encode output to the process. For a list of possible coding systems, use M-x list-coding-systems. |
| Visit current buffer's file with specified coding system | C-x RET r | (revert-buffer-with-coding- system CODING-SYSTEM &optional FORCE) | Visit the current buffer's file again using coding system CODING-SYSTEM. • For a list of possible values of CODING-SYSTEM, use M-x list-coding-systems. • If CODING-SYSTEM leaves the text conversion unspecified, or if it leaves the end-of-line conversion unspecified, FORCE controls what to do. • If FORCE is nil, get the unspecified aspect (or aspects) from the buffer's previous 'buffer-file-coding-system' value (if it is specified there). Otherwise, determine it from the file contents as usual for visiting a file. |
| Set coding system of terminal output | C-x RET t | (set-terminal-coding- system CODING-SYSTEM &optional TERMINAL) | Set coding system of terminal output to CODING-SYSTEM. All text output to TERMINAL will be encoded with the specified coding system. For a list of possible values of CODING-SYSTEM, use M-x list-coding-systems. The default is determined by the selected language environment or by the previous use of this command. TERMINAL may be a terminal object, a frame, or nil for the selected frame's terminal. The setting has no effect on graphical terminals. |
| Select coding system to communicate with other X clients | C-x RET x | (set-selection-coding- system CODING-SYSTEM) | Make CODING-SYSTEM used for communicating with other X clients. • When sending or receiving text via cut_buffer, selection, and clipboard, the text is encoded or decoded by CODING-SYSTEM. |
| Display all coding categories | M-: (list- coding- categories) | (list-coding-categories) | Display a list of all coding categories. This is not an interactive function (a command). Therefore you must execute inside a Emacs Lisp program or interactively by evaluating the (list-coding-categories) form using M-: |
| Display all coding systems | | (list-coding-systems &optional ARG) | Display a list of all coding systems. This shows the mnemonic letter, name, and description of each coding system. With prefix ARG, the output format gets more cryptic, but still contains full information about each coding system. |

Input Method — References

| Topic & link | Description | |
|---|---|--|
| Input Method | | |
| GNU Emacs Manual: International - Selecting an Input Method | Manual: Introduction, general concepts | |
| GNU Emacs Manual: Basic - Inserting Text | Manual: Describes C-q concepts, C-x 8 concepts and some other ones. | |
| GNU Emacs Manual: Text - Quotation Marks | | |
| Mastering Emacs - Olé! Diacritics in Emacs | Mickey Petersen's article on how to type diacritic characters. | |
| Wikipedia - Compose key | General description of the concept of character/key composition. | |
| How to enter Greek characters in Emacs @ Stack Overflow | An interesting set of various ideas to control how to enter those characters. | |
| Change prefix to compose character @ Stack Overflow | | |
| File Encoding | | |

| Topic & link | Description |
|--|--|
| Emacs Manual - Coding Systems | |
| Emacs File Encoding FAQ @ ErgoEmacs | |
| Changing Encodings @ Emacs Wiki | |
| How to see the file's encoding in Emacs @ Stack Overflow | |
| | |
| Encoding Techniques/Standards | |
| The differences between ASCII, ISO 8859, and Unicode @ Indiana University | Quick introduction |
| ISO/IEC 8859 @ Wikipedia | Includes several parts (8 bit mappings), including: • Part 1: ISO/IEC 8859-1 (latin-1, Western European) • Part 2: ISO/IEC 8859-2 (latin-2, Central European • up to part 16 (see main wikipedia page, the table contains links to the information on all parts). |
| ASCII / ISO 8859-1 (Latin-1) Table with HTML Entity Names @ Standford University | |
| • Unicode | |
| Unicode @ Wikipedia | |
| Lists of Unicode characters @ Wikipedia | |
| Unicode Characters and Properties @ RegularExpression Info | Unicode and characters with accents (or marks) can be encoded in several ways. This complicates searching and you must be aware of this to support more than just English. |