

Operation	Keystroke	Function		Note
Insert Greek Letter <ul style="list-style-type: none"> With the <f6> g prefix <p>See also: Inserting Text</p>	<f6> g	Insert a greek letter by typing <f6> g followed by a key in [a-zA-Z] range inserts the Unicode character for the equivalent Greek letter. <ul style="list-style-type: none"> This PEL key binding is always available. Examples: <p><f6> g a inserts α <f6> g b inserts β <f6> g A inserts Α</p> <p><f6> g B inserts Β <f6> g 1 inserts λ</p> <ul style="list-style-type: none"> The insertions work everywhere insert is allowed, including in response to prompts. To see all keys: <ul style="list-style-type: none"> Type <f6> g C-h Use which-key mode and just type <f6> g 		
Insert Greek Letter <ul style="list-style-type: none"> With <f9> key: <ul style="list-style-type: none"> Available only when the pel-activate-f9-for-greek user-option is turned on. <p>See also: Inserting Text</p>	<f9>	Insert a greek letter by typing <f9> followed by a key in [a-zA-Z] range inserts the Unicode character for the equivalent Greek letter. <ul style="list-style-type: none"> Only available when pel-activate-f9-for-greek user-option is turned on. Examples: <p><f9> a inserts α <f9> b inserts β <f9> A inserts Α <f9> B inserts Β <f9> 1 inserts λ</p> <ul style="list-style-type: none"> The insertions work everywhere insert is allowed, including in response to prompts. To see all keys: <ul style="list-style-type: none"> Type <f9> C-h Use which-key mode and just type <f9> This is not a command bound to a key: it's an additional set of bindings added to Emacs key-translation-map .		
Start pel-Σgreek Hydra <ul style="list-style-type: none"> Quickly type succession of Greek characters <p>See also: Inserting Text</p>	<f7> <f6> <f6>	Start the Greek letter Hydra . <ul style="list-style-type: none"> After typing <f7> <f6> <f6> type one of the Meta letter keys in the hydra to insert a Greek character. Type any other character to insert them, latin letters, digits, punctuation characters, and Meta-char to inter the greek character, etc... In terminal mode the cursor keys may not work because they are often encoded using Esc keys with is mapped to Meta. Requires the hydra external package. activated by pel-use-hydra . You must also set pel-activate-hydra-for-greek to t to activate this hydra. <ul style="list-style-type: none"> Exit the hydra by typing <f7> <p>With this Hydra, you can type Greek text by pressing the Meta keys. Με αυτό το Hydra, μπορείτε να πληκτρολογήσετε ελληνικό κείμενο πατώντας τα πλήκτρα Meta.</p> <p>No prefix is active: all digit keys can be typed: 0123456789 ;; M-u for undo is not available: use ``<f11> u u`` instead. ;; In terminal mode the cursor keys may not work though. Use C-b, C-f, C-n, C-p instead. Exit the Hydra by typing <f7></p> <pre>--UUU:*--F1 *scratch* All (9,29) (Lisp Interaction s* WK Anzu Fly ^ ElDoc) 12:56pm ----- [M-a]: α, [M-b]: β, [M-c]: γ, [M-d]: δ, [M-e]: ε, [M-f]: φ, [M-g]: υ, [M-h]: η, [M-i]: ι, [M-j]: ϕ, [M-k]: κ, [M-l]: λ, [M-m]: μ, [M-n]: ν, [M-o]: ο, [M-p]: π, [M-q]: θ, [M-r]: ρ, [M-s]: σ, [M-t]: τ, [M-u]: υ, [M-w]: ω, [M-x]: ξ, [M-y]: ψ, [M-z]: ζ, [M-A]: Α, [M-B]: Β, [M-C]: Χ, [M-D]: Δ, [M-E]: Ε, [M-F M-J]: Φ, [M-G]: Γ, [M-H]: Η, [M-I]: Ι, [M-K]: Κ, [M-L]: Λ, [M-M]: Μ, [M-N]: Ν, [M-O]: Ο, [M-P]: Π, [M-Q]: Θ, [M-R]: Ρ, [M-S]: Σ, [M-T]: Τ, [M-U]: Υ, [M-W]: Ω, [M-X]: Ξ, [M-Y]: Ψ, [M-Z]: Ζ, [<f7>]: cancel.</pre>		
Some of the special characters that can be inserted with C-x 8 keys	C-x 8 ! C-x 8 \$ C-x 8 + C-x 8 - C-x 8 . C-x 8 < C-x 8 = C-x 8 > C-x 8 ? C-x 8 C C-x 8 L C-x 8 P C-x 8 R C-x 8 S C-x 8 Y C-x 8 c C-x 8 o	! \$ ± - . « ~ » ¿ © £ ¶ ® § ¥ ¢ °	C-x 8 - A C-x 8 - D C-x 8 - N C-x 8 - O C-x 8 - T C-x 8 - a C-x 8 - n C-x 8 - o C-x 8 - t C-x 8 - ~ C-x 8 ` A C-x 8 ` E C-x 8 ` O C-x 8 ` U C-x 8 ` a C-x 8 ` e C-x 8 ` i	Ã Ð Ñ Õ Þ ã ñ õ þ ¬ À È Ò Ù à è ì
Some of the special characters that can be inserted with C-x 8 keys (continued).	C-x 8 u C-x 8 x C-x 8 a < C-x 8 a = C-x 8 a > C-x 8 N o C-x 8 2 + C-x 8 ~ SPC C-x 8 ~ = C-x 8 / / C-x 8 / = C-x 8 3 / 4 C-x 8 1 / 2 C-x 8 1 / 4 C-x 8 ^ 1 C-x 8 ^ 2 C-x 8 ^ 3 C-x 8 * *	μ × ← ↔ → ∞ ‡ ~ ≈ + ≠ ¾ ½ ¼ ¹ ² ³ •	C-x 8 ` o C-x 8 ` u C-x 8 _ < C-x 8 _ > C-x 8 _ a C-x 8 _ o C-x 8 ^ a C-x 8 ^ e C-x 8 " a C-x 8 " e C-x 8 " i C-x 8 " o C-x 8 " s C-x 8 " u C-x 8 , C C-x 8 , c C-x 8 / e C-x 8 / o	ò ù ≤ ≥ ä ø â ê ä ë ï ö ß ü Ç ç æ ø
File Encoding	The following commands and functions can help investigate and handle file encoding issues.			
Show encoding of file visited in current buffer <ul style="list-style-type: none"> See also: Help/Info 	<f11> ? d e	(pel-show-buffer-file-encoding)	Show coding system of file in current buffer. <ul style="list-style-type: none"> Open a "Help" buffer and show the value of the buffer-file-coding-system variable. 	
Query info about point <ul style="list-style-type: none"> See also: Help/Info 	<ul style="list-style-type: none"> C-x = <f11> ? d p 	(what-cursor-position &optional DETAIL)	Displays information about character at point in the echo area: position, character, encoding. <ul style="list-style-type: none"> With any prefix argument opens a "Help" buffer and show the complete information of character at point with all properties, face, encoding, etc. <ul style="list-style-type: none"> Type: C-u C-x = With PEL, you can also type: C-- C-x = 	

Operation	Keystroke	Function	Note
Visit a file literally: with no encoding support and conversion See also: 🔗 File-mngt	<f11> f M-1	(find-file-literally FILENAME)	Visit file FILENAME with no conversion of any kind. <ul style="list-style-type: none"> 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.
Change text encoding 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. 🍌 To change line-endings of a file, use on of the following: <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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 l	(set-language-environment LANGUAGE-NAME)	Set up multilingual environment for using LANGUAGE-NAME. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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 <i>diacritic</i> 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: <ul style="list-style-type: none"> 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	
<ul style="list-style-type: none"> 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.