↖ Topic Index

Insert Control/Unicode Characters - Input Method - File encoding

Operation	Keystroke	Function	Note
Input Methods & File Encoding ○ Input method help, ASCII chart, • Insert quoted, accentuated letter • Insert char by name ○ Insert Greek characters • Insert special characters ○ File encoding • query encoding, list all encodings • set/restore file encoding • change line ending	This PDF describes: • **Emacs input methods**, how to type characters from various character sets, • **Emacs file encoding**, the way Emasc stores the text into file using various **character encoding systems**. PEL supports the following external package: 📦 The **ascii-table** external package that PEL activates when the **pel-use-ascii-table** user option is set to **t**. 📦 The **hydra** external package that PEL activates activated when the **pel-use-hydra** user option is set to **t.** Used for pel-∑greek **Hydra** More background information is available in the **Reference Table section**.		
Open this PDF file. See also: **Σ Help/Info**	`<f11> t <f1> 2`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the **Σ Input Method** 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: **Σ Key-Chords**	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 ✂ Emacs default key for this is: `C-\` . PEL rebinds it!!	`<f11> t i`	(**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`	(**set-input-method** INPUT-METHOD &optional INTERACTIVE)	Set the input method for the current buffer only. • Always prompts.
Get Help on Input Method ☝ **Lists all sequences → characters**	• `C-h I` • `C-h C-\` • `<f1> I`	(**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`	(**list-input-methods**)	Lists all input methods supported. Like french-prefix, TeX, IPA, etc…
Describe encoding system **Describe buffers encoding ➡** See also: **Σ Help/Info**	• `C-h C` • `<f1> C` • `<f11> ? d C`	(**describe-coding-system** CODING-SYSTEM)	Display information about CODING-SYSTEM. • Prompts for coding system name. Supports completion. • ☝ Type RET to describe current buffer encoding.
Describe language environment See also: **Σ Help/Info**	• `C-h L` • `<f1> L`	(**describe-language-environment** LANGUAGE-NAME)	Describe how Emacs supports language environment LANGUAGE-NAME. • Prompts for language name, proposing currently used as default. Supports completion.
Query info about point **Show information about current character.** See also: **Σ Help/Info**	• `C-x =` • `<f11> ? d p`	(**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 =`
	`<f11> ? d P`	(**pel-what-cursor-position**)	Same as above but always display the complete information.
Display ASCII table See also: **Σ Help/Info**	`<f11> ? A`	(**ascii-table**)	Show an interactive ASCII table in the other (next) window. 📦 Requires the **ascii-table** package that PEL activates this when the **pel-use-ascii-table** 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.		
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	The same command can be used to enter accented letters (although specialized input methods can also be used, like French input method or other latin or European input methods (see `C-h I` above). The following are just **examples** showing the accent selectors.		
Insert accent: grave	`C-x 8 ` e`	(**self-insert-command** N)	Inserts: **è**
Insert accent: acute	`C-x 8 ' e`	(**self-insert-command** N)	Inserts: **é**
Insert accent: circumflex	`C-x 8 ^ e`	(**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	The following commands allow you to enter Unicode characters by name or by value at point.		
Insert arbitrary character by name	`C-x 8 RET`	(**insert-char** CHARACTER &optional COUNT INHERIT)	Insert COUNT copies of CHARACTER. • COUNT is given by the prefix argument. If omitted, it defaults to 1.
Insert Unicode character by name ➡ **Insert Unicode character by value ➡** In macOS you may also want to: add Greek as a Keyboard Input Sources and temporary select it to enter Greek characters directly.	Prompts for CHARACTER. You can specify CHARACTER: • as its **Unicode character name**, 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), • as a **code point with a radix specified with #**, e.g. #o21430 (octal), #x2318 (hex), or #10r8984 (decimal). **To get a list of all supported Unicode characters**, type: `C-x 8 RET <tab> <tab>` **Example**: to insert the Lambda greek letter λ you can type one of these: • `C-x 8 <RET> 03bb <RET>` • `C-x 8 <RET> GREEK SMALL LETTER LAMBDA <RET>` • To see the available choices using tab completion, type: `C-x 8 <RET> * LAMBDA <tab>`		

Operation	Keystroke	Function	Note

Insert Greek Letters:
- **using TeX input method in any OS**

List all TeX char sequences with **`<f1> I TeX`**

To insert greek letters or other mathematical symbols, use the TeX input method (`<f11> t I TeX` or `C-u <f11> t i TeX`) .
- For example the following keystrokes produce Greek letters and other symbols.

`\Alpha`	Α	`^\beta`	ᵝ
`\Beta`	Β	`\Stigma`	Ϛ
`\Delta`	Δ	`\sum`	∑
`\lambda`	λ	`\forall`	∀
`\gamma`	γ	`\frac25`	⅖

Insert Greek Letter

- With the **`<f6> g`** prefix

See also: **Inserting Text**

Keystroke: `<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.
- This PEL key binding is always available.

Examples:

`<f6> g a` inserts **α** `<f6> g b` inserts **β** `<f6> g A` inserts **Α**
`<f6> g B` inserts **Β** `<f6> g l` inserts **λ**

- The insertions work everywhere insert is allowed, including in response to prompts.
- To see all keys:
  - Type `<f6> g C-h`
  - Use which-key mode and just type `<f6> g`

Insert Greek Letter
- With **`<f9>`** key:
  - Available only when the **pel-activate-f9-for-greek** user-option is turned on.

See also: **Inserting Text**

Keystroke: `<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.
- Only available when **pel-activate-f9-for-greek** user-option is turned on.

Examples:

`<f9> a` inserts **α** `<f9> b` inserts **β** `<f9> A` inserts **Α** `<f9> B` inserts **Β** `<f9> l` inserts **λ**

- The insertions work everywhere insert is allowed, including in response to prompts.
- To see all keys:
  - 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-∑greek Hydra
- Quickly type succession of Greek characters

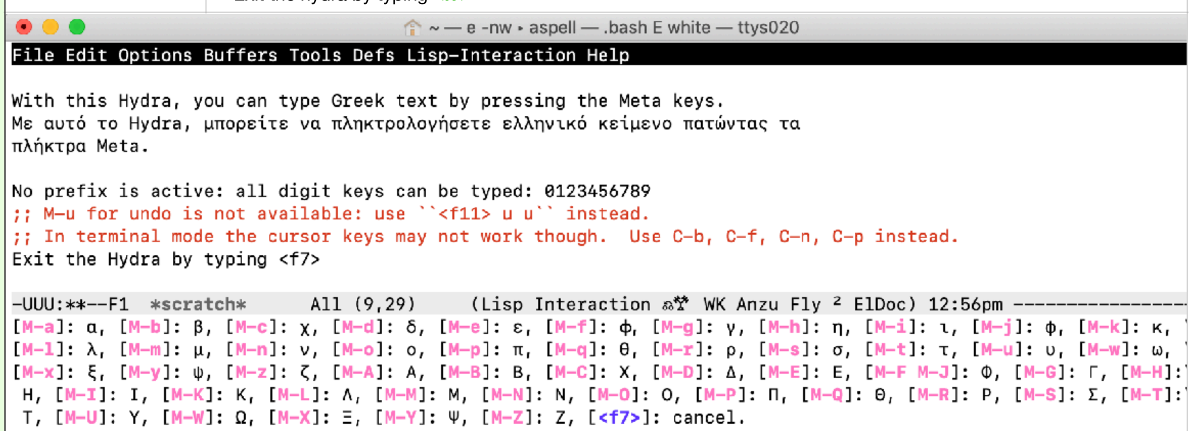
See also: **Inserting Text**

Keystroke: `<f7> <f6> <f6>`

Start the Greek letter **Hydra**.
- 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.
- Exit the hydra by typing `<f7>`



```
~ — e -nw ▸ aspell — .bash E white — ttys020
File Edit Options Buffers Tools Defs Lisp-Interaction Help

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.

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>

-UUU:**--F1  *scratch*     All (9,29)     (Lisp Interaction ஃ✲ WK Anzu Fly ² ElDoc) 12:56pm ----------------
[M-a]: α, [M-b]: β, [M-c]: χ, [M-d]: δ, [M-e]: ε, [M-f]: ф, [M-g]: γ, [M-h]: η, [M-i]: ι, [M-j]: ф, [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.
```

Some of the special characters that can be inserted with C-x 8 keys

`C-x 8 !`	¡	`C-x 8 ~ A`	Ã
`C-x 8 $`	¤	`C-x 8 ~ D`	Ð
`C-x 8 +`	±	`C-x 8 ~ N`	Ñ
`C-x 8 -`	-	`C-x 8 ~ O`	Õ
`C-x 8 .`	·	`C-x 8 ~ T`	Þ
`C-x 8 <`	«	`C-x 8 ~ a`	ã
`C-x 8 =`	¯	`C-x 8 ~ n`	ñ
`C-x 8 >`	»	`C-x 8 ~ o`	õ
`C-x 8 ?`	¿	`C-x 8 ~ t`	þ
`C-x 8 C`	©	`C-x 8 ~ ~`	¬
`C-x 8 L`	£	``C-x 8 ` A``	À
`C-x 8 P`	¶	``C-x 8 ` E``	È
`C-x 8 R`	®	``C-x 8 ` O``	Ò
`C-x 8 S`	§	``C-x 8 ` U``	Ù
`C-x 8 Y`	¥	``C-x 8 ` a``	à
`C-x 8 c`	¢	``C-x 8 ` e``	è
`C-x 8 o`	°	``C-x 8 ` i``	ì

Some of the special characters that can be inserted with C-x 8 keys (continued).

`C-x 8 u`	µ	``C-x 8 ` o``	ò
`C-x 8 x`	×	``C-x 8 ` u``	ù
`C-x 8 a <`	←	`C-x 8 _ <`	≤
`C-x 8 a =`	↔	`C-x 8 _ >`	≥
`C-x 8 a >`	→	`C-x 8 _ a`	ª
`C-x 8 N o`	№	`C-x 8 _ o`	º
`C-x 8 2 +`	‡	`C-x 8 ^ a`	â
`C-x 8 ~ SPC`	~	`C-x 8 ^ e`	ê
`C-x 8 ~ =`	≈	`C-x 8 " a`	ä
`C-x 8 / /`	÷	`C-x 8 " e`	ë
`C-x 8 / =`	≠	`C-x 8 " i`	ï
`C-x 8 3 / 4`	¾	`C-x 8 " o`	ö
`C-x 8 1 / 2`	½	`C-x 8 " s`	ß
`C-x 8 1 / 4`	¼	`C-x 8 " u`	ü
`C-x 8 ^ 1`	¹	`C-x 8 , C`	Ç
`C-x 8 ^ 2`	²	`C-x 8 , c`	ç
`C-x 8 ^ 3`	³	`C-x 8 / e`	æ
`C-x 8 * *`	•	`C-x 8 / o`	ø

Operation	Keystroke	Function	Note
File Encoding	The following commands and functions can help investigate and handle file encoding issues. See: • **Emacs Coding System** , from Emacs Manual • **Working with Coding Systems and Unicode in Emacs**, from Mastering Emacs , by Mickey Petersen		
Show encoding of file visited in current buffer • See also: **Σ Help/Info**	`<f11> ? d e`	(**pel-show-buffer-file-encoding**)	Show coding system of file in current buffer. • Open a *Help* buffer and show the value of the buffer-file-coding-system variable.

Query info about point
• See also: **Σ Help/Info**
	• `C-x =`
	• `<f11> ? d p`
		(**what-cursor-position** &optional DETAIL)	Displays information about character at point in the echo area: position, character, encoding.
			• ☝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 all coding systems	`<f11> f M-c`	(**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.
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-:`
Visit a file literally: with no encoding support and conversion See also: **Σ File-mngt**	`<f11> f M-l`	(**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.
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. • 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. ☝**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.
Setup multilingual environment	`C-x RET l`	(**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.
Specify files coding system • **Use to switch back buffer's coding**	`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.

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	
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.