

Auto-Completion Support

Description	Keystroke	Function	Note
Auto Completion <ul style="list-style-type: none"> Help @ Customization Display auto-completion info Select auto-completion mode Built-in completion completion-preview-mode External completion modes <ul style="list-style-type: none"> auto-complete company-mode corfu Auto-completion Reference 	Emacs provides support for text completion, called auto-completion . <ul style="list-style-type: none"> Emacs provides a built-in auto-completion system, accessible via the (completion-at-point) command bound to C-M-i . PEL supports the following external packages that provide alternative completion at point with pop-up menu: 		
	 auto-complete	 pel-use-auto-complete	The oldest external completion engine.
	 company-mode	 pel-use-company	A powerful completion engine.
	 corfu	 pel-use-corfu	Use CO mpletion in R egion FU nction. A very powerful and popular auto-completion engine. Available for Emacs >= 29.1 only.
	 corfu-terminal	 pel-use-corfu-terminal	Use corfu popup in terminal Emacs < 31. PEL activates it when pel-use-corfu is on, but Emacs < 31 and runs in terminal mode.
	For these PEL supported modes, you can access the customization buffer quickly with the <f11> c <f2> key sequence.		
	To activate an automatic completion mode for a major mode add one of the following special PEL functions to the list of minor modes listed in the major mode specific instance of:		
	 pel-MODE-activates-minor-modes		Add one of the following special PEL functions that activate the corresponding engines: <ul style="list-style-type: none"> pel-auto-complete-mode, pel-company-mode, pel-corfu-mode
	<ul style="list-style-type: none"> For example, add completion-preview-mode and pel-company-mode to  pel-elisp-activates-minor-modes to automatically activate these in all Emacs Lisp buffers. Emacs >= 30 provides the completion-preview-mode which provides an easy-to-use in-buffer completion help that complements the above. 		
	 More on abbreviation completion is available in the Abbreviations table. Both abbreviation completion and one auto-completion mechanism can be used at the same time (using different keys if any), in some case the abbreviation choices can also be available via auto-completion.  More dynamic completion modes based on Eglot or LSP are supported by Emacs but not yet documented in PEL.		
Last updated on:	2025-12-19		
Open this PDF file. See also: ℹ Help/Info	<f11> c <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the ℹ Auto-Completion 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.
Customize PEL auto-completion support. See also: ℹ Customize	<f11> c <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Open the PEL customize group(s) for the current context: auto-completion support. Use this to open to change PEL user option variables the activate and control the various Apple script features such as the name of the narrator voice. <ul style="list-style-type: none"> When a prefix argument (like C-u) opens the buffer inside another window.
Customize Emacs built-in auto-completion support See also: ℹ Customize	<f11> c <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs auto-completion group which includes: auto-complete, company, hippie-expand. When prefix arg. (like C-u) opens the buffer inside another window.
	 Group belonging to files that have not yet been loaded are normally not accessible in Emacs and via the customize-group command. PEL, however, attempts to locate the file that defines a non-loaded customization group and will prompt you for loading the file if it finds it.		
Display Auto-completion info: • status and configuration  Provides quick access buttons to change customizable user-options.	<f11> c ?	(pel-completion-info &optional APPEND)	Print information about available auto-completion info in a "pel-autocomplete-info" help-mode buffer. Clear previous buffer content unless a prefix arg (like C-u) is used.
	<ul style="list-style-type: none"> Prints current state and values of relevant user-options as buttons you can use to get more info and change their customized values. Shows which one is enabled via customization and their current activation state.  Underlines links are buttons that open the customization buffer where you can change the customized value. 		
Select auto-completion mode	<f11> c =	(pel-select-auto-complete-tool &GLOBALLY)	Prompt user to select auto-completetion mode to use in current buffer. <ul style="list-style-type: none"> With prefix arg (such as C-u) select for all buffers. Select: Emacs built-in auto-completion, auto-complete , company-mode , corfu .
Emacs Built-in Completion	Emacs built-in completion is provided by the completion-at-point command, described below. <ul style="list-style-type: none"> This automatic text completion is always available but can be extended or overridden by mechanisms used by the external auto-completion packages. The pel-completion-info command prints the values of the various variables and user-options controlling its behaviour. 		
Symbol Completion at point See also: ℹ Xref 	<ul style="list-style-type: none"> C-M-i <Esc> <tab> M-<tab> 	(completion-at-point)	Perform completion on the text around point. <ul style="list-style-type: none"> The completion method is determined by 'completion-at-point-functions'. The tags-completion-at-point-function is used for Emacs Lisp code by default. <ul style="list-style-type: none"> It provides a list of possible values in the "Completions" buffer. The key bindings to select from "completions" buffer are shown below.
	C-M-i is also used for Flyspell, which can be used to spell check only moments and strings.		
Symbol Completion at point • and language specific completion	<f6> c	(complete-symbol ARG)	Perform completion of the text around point, using the method identified by the variable completion-at-point-functions , acting as the completion-at-point command above. <ul style="list-style-type: none"> With prefix argument, such as C-u, this does completion within the collection of symbols listed in the index of the manual for the language you are using.
Complete language specific symbol at point	<f6> C	(info-complete-symbol &optional MODE)	Perform completion of symbol at point using mode-specific language items. <ul style="list-style-type: none"> For example, inside a Emacs-Lisp buffer it finds the Emacs-Lisp functions, variables names.
Completion function - using tags Candidate for: completion-at-point-functions		(tags-completion-at-point-function)	Using tags, return a completion table for the text around point. If no tags table is loaded, do nothing and return nil. This uses the tag facility.
Toggle Completion Preview mode Emacs >= 30			
A minor mode that shows possible completion in a light face. Can be available while another completion major-mode is used.  The following keys are available whenever the "Completions" buffer shows a list of potential completions. The keys are typed at point, there's no need to switch to the window of the "Completion" buffer.  These are available for completion preview but also for completion-at-point .	<f11> c p	(completion-preview-mode &optional ARG)	Toggle completion-preview-mode in current buffer. <ul style="list-style-type: none"> Activate it with positive prefix argument, disable it with negative prefix argument.
	<f11> c P	(global-completion-preview-mode &optional ARG)	Toggle Completion-Preview mode in all buffers. <ul style="list-style-type: none"> Activate it with positive prefix argument, disable it with negative prefix argument.
	<tab>	Expand suggested completion in buffer. Type more characters to further refine the search.	
	M-i	Open a "Completion" buffer listing all possible completion candidates. Does not move point. <ul style="list-style-type: none"> Select other candidate show in the "Completion" buffer with M-<up> and M-<down>. Then chose one with M-RET. 	
	M-<down>	(minibuffer-next-completion &optional N VERTICAL)	Selects and show next completion candidate. <ul style="list-style-type: none"> Move by N positions with numeric prefix N
	M-<up>	(minibuffer-previous-completion &optional N)	Selects and show previous completion candidate. <ul style="list-style-type: none"> Move by N positions with numeric prefix N
	M-RET	(minibuffer-choose-completion &optional NO-EXIT NO-QUIT)	Select candidate. Run 'choose-completion' from the minibuffer in its completions window. <ul style="list-style-type: none"> With prefix argument NO-EXIT, insert the completion candidate at point to the minibuffer, but don't exit the minibuffer. When the prefix argument is not provided, then whether to exit the minibuffer depends on the value of 'completion-no-auto-exit'. If NO-QUIT is non-nil, insert the completion candidate at point to the minibuffer, but don't quit the completions window.

Auto-completion — References

Document	Note
Basic Auto Completion	GNU Emacs Manual - Completion for Symbol Names
Auto Completion with Auto-Complete	
Auto Complete @ MELPA	You can get auto-complete from MELPA. An interesting point of this page lists the other packages that need auto-complete. There's over 45 packages that use it for various programming languages and environments.
Auto Complete @ GitHub	Auto complete source code
Auto Complete Manual @ Github	Covers installation, check, features, concepts, configuration, advanced usage. Reading required for users.
Using Emacs: 8 - Auto-complete @ Youtube	Mike Zamansky video that covers abbreviation and auto-complete. Duration: 5 minutes.
Using Emacs: 45- Company or Autocomplete @ Youtube	Another video from Mike Zamansky that covers both auto-complete and company-mode. Duration: 13 minutes.
Auto Completion with Company-mode	
company-mode : Modular in-buffer completion framework for Emacs	Text completion framework for emacs
Using digits to select company-mode candidates @ (or emacs irrelevant)	