



Lispy minor mode🚧

Description	Key	Function	Note
Lispy : Context-based modal editing of Lisp code	<p>The lispy mode provides modal editing to Emacs for Lisp-like languages.</p> <ul style="list-style-type: none">Lisp is a very structured programming language, made of succession and combinations of S-expressions (“sexp”): lists that start with (and end with) “paren”. Lispy takes advantage of the structure of Lisp code.As long as point (the cursor) is before the left, opening, paren or after the right, closing paren, the keys are interpreted as lispy commands.<ul style="list-style-type: none">Keys in other locations are interpreted as usual. <p>This table lists the lispy command keys, with links to the Lispy function Reference for each one.</p> <p>📦 This requires the lispy external package. 📦 PEL downloads, installs and activates lispy when the pel-use-lispy user option is set to t.</p> <p>👤👉 To get lispy mode run when Emacs visits a file of a specified mode, include the major mode in the PEL user-option pel-modes-activating-lispy .<ul style="list-style-type: none">PEL does not activate lispy for any major mode by default. That’s OK to learn lispy by activating it for testing. But once you learn and are comfortable with it you will want to activate when the file is opened automatically by adding the major mode in that list.</p>		
🔗 Customize PEL use of Lispy and Lispy itself.	<f11> <f2> SPC M-L	(pel-cfg-pkg-lisp &optional OTHER-WINDOW)	Prompt to customize: <ol style="list-style-type: none">PEL lispy support for Emacs Lisp and Common Lisplispy itself. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
Toggle Lispy mode See also: <ul style="list-style-type: none">🔗 - Common Lisp🔗 - Emacs Lisp	<ul style="list-style-type: none"><f12> M-L<M-f12> M-L <f11> SPC 1 M-L	(pel-lispy-mode &optional ARG)	Toggle lispy-mode on/off. Lispy is a minor mode for navigating and editing LISP dialects. 📦 Requires lispy external package . 📦 PEL downloads, installs and configure it when pel-use-lispy user option is set to t . Please read the information on lispy web site . 🖥️ pel-lispy-mode calls lispy-mode but also prepares hydra, loaded dynamically with PEL. 👉 Set the pel-modes-activating-lispy user-option to activate lispy automatically for major modes.
Getting Code Help			
Describe function at point See Also: 🔗 Help/Info	<ul style="list-style-type: none">C-1 <		

Description	Key	Function	Note																
<u>View: center current sexp</u>	v	(special-lispy-view)	Recenter current sexp to be on the first line of the window. When called twice in a row, recenter back to the original position.																
<u>Visit another file</u> See: ☞ Projectile	V	(special-lispy-visit ARG)	Visit another file within this project using projectile or find-file-in-project . <ul style="list-style-type: none">Customize lispy-visit-method to select what function to use.<ul style="list-style-type: none">  PEL supports both of these external packages, and use the pel-use-projectile and pel-use-find-file-in-project user-options to download and activate each one. Unless you are familiar with find-file-in-project you may find projectile more useful and faster.Use V to open the file in the current window.Use 2V to open the file in another window.																
Search																			
<u>Occur search inside the current top-level sexp</u>	y	(special-lispy-occur)	Do an occur for the current top-level sexp. Go back-to-paren afterwards. This is useful e.g. to see where a particular variable is used within the current defun.																
Goto Definition																			
<u>goto definition</u> using directory tabgs	g	(special-lispy-goto &optional ARG)	Jump to symbol within files in current directory . Prompt for symbol and jump to it. <ul style="list-style-type: none">When ARG isn't nil, call 'lispy-goto-projectile' instead.See lispy goto wiki page.																
<u>goto definition</u> using projectile base directory	<ul style="list-style-type: none">Ogogp	(lispy-goto-projectile)	Jump to symbol within files in ('projectile-project-root').																
<u>goto definition in local file</u>	G	(special-lispy-goto-local &optional ARG)	Similar to lispy-goto, but only current file's tags are used instead of whole directory's tags.																
<u>Follow: jump to definition</u>	<ul style="list-style-type: none">F	(special-lispy-follow)	When region is active jump to the definition of marked symbol. Otherwise jump to the definition of the first symbol in current sexp.																
	<ul style="list-style-type: none">M- .	(lispy-goto-symbol SYMBOL)																	
<u>Pop tag</u>	<ul style="list-style-type: none">D	(special-pop-tag-mark)	Go back from where it came with Follow																
	<ul style="list-style-type: none">M- ,	(pop-tag-mark)																	
Narrow/Widening See also: ☞ Narrowing	<ul style="list-style-type: none">Narrowing hides everything in the buffer except the selected region, allowing work on that region alone.Widen it back to see the complete buffer again.																		
<u>Narrow current sexp region</u>	N	(special-lispy-narrow ARG)	Narrow current sexp or region.																
<u>Widen</u>	W	(special-lispy-widen)	Widen back to see the complete buffer.																
Cut/Paste/Mark/Hide/Indent																			
<u>Indent / hide/show outline</u>	i	<ul style="list-style-type: none">With no active region: (special-lispy-tab)	If in outline: hide/show outline, otherwise indent all code of current paren <ul style="list-style-type: none">When region is active, call 'lispy-mark-car'.																
<u>mark car</u>	i	<ul style="list-style-type: none">With active region: (lispy-mark-car)	Mark the car of current thing. <ul style="list-style-type: none">Updates lispy-back history.																
<u>Copy region or sexp to kill ring</u>	n	(special-lispy-new-copy)	Copy marked region or sexp to kill ring.																
<u>Mark list</u>	m	(special-lispy-mark-list ARG)	Mark the current sexp. If mark is already active, deactivate it instead. When ARG is more than 1, mark ARGth element. <ul style="list-style-type: none">Updates lispy-back history.																
<u>Paste</u>	P		When region is active, replace it with current kill. Forward to yank otherwise.																
Edit code	Transform code using the following commands																		
<u>clone</u>	c	(special-lispy-clone ARG)	Clone sexp ARG times. <ul style="list-style-type: none">When the sexp is top level, insert an additional newline.																
<u>Toggle to the other mode</u>	o	(special-lispy-other-mode &optional ARG)	Prefix to the following other verbs: <table><tr><td>key</td><td>binding</td></tr><tr><td>---</td><td>-----</td></tr><tr><td>SPC</td><td>special-lispy-other-space</td></tr><tr><td>g</td><td>special-lispy-goto-mode</td></tr><tr><td>h</td><td>special-lispy-move-left</td></tr><tr><td>j</td><td>special-lispy-down-slurp</td></tr><tr><td>k</td><td>special-lispy-up-slurp</td></tr><tr><td>l</td><td>special-lispy-move-right</td></tr></table>	key	binding	---	-----	SPC	special-lispy-other-space	g	special-lispy-goto-mode	h	special-lispy-move-left	j	special-lispy-down-slurp	k	special-lispy-up-slurp	l	special-lispy-move-right
key	binding																		
---	-----																		
SPC	special-lispy-other-space																		
g	special-lispy-goto-mode																		
h	special-lispy-move-left																		
j	special-lispy-down-slurp																		
k	special-lispy-up-slurp																		
l	special-lispy-move-right																		
<u>Move current sexp to the left</u>	oh																		
<u>Move current sexp inside first element of list below</u>	oj																		
<u>Move current sexp to become last element of list above</u>	ok																		
<u>Move current sexp to the right, outside current list</u>	ol																		
<u>Raise: use current sexp as replacement for its parent</u>	r																		
<u>Raise: current and next previous sexp as replacement for their parent</u>	R		Use current sexp and the following (if called from the left), or the preceeding (if called from the right) sexps, or the active region as replacement for their parent.																
<u>Move sexp down in list</u>	s		Move current sexp or region down arg times. Don't exit the parent list. Also works for outlines.																
<u>undo</u>	u																		
<u>Move current sexp up</u>	w		Move current sexp or region up arg times. Don't exit the parent list. Also works for outlines.																
<u>Bind var: current sexp to let bound variable</u>	xb		Transform the current list expression into a let-bound variable; iedit-mode is used to name the new variable. Use M-m to finish naming the variable.																
<u>Unbind a let bound variable</u>	xu		Unbind a let-bound variable. Also works for Clojure.																
<u>turn current lambda into a defun</u>	xd																		

Description	Key	Function	Note
turn current defun into a lambda	x l		
turn nested if into cond	x c		
turn cond into nested if expressions	x i		
Inline current function or macro call	x f		Inline current function or macro call, i.e. replace it with function body. The function should be interned and its body find-able.
Convolute: Exchange the order of application of 2 closest outer forms	C		Exchange the order of application of two closest outer forms, relative to current expression or region.
Slurp: grow either current sexp or region	>		Grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-barf. <ul style="list-style-type: none"> With an arg of 0, grow as far as possible. With an arg of -1, grow until the end of the line where the current sexp ends or as far as possible before that position.
Barf: shrink either current sexp or region	<		Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-slurp.
Splice the current list into the parent list	/		Splice the current list into the parent list. Move the point to the next list to splice in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction.
Move to beginning of current defun	A		Forward to beginning-of-defun. When called twice in a row, restore the previous point and mark positions.
Teleport: move current sexp to Ace target	t		Move current sexp to Ace target inside current function
	tt		Move current sexp to Ace target to any sexp inside current window
Move to Ace target symbol & erase to replace	H		Calls lispy-ace-symbol and deletes the selected symbol.
Convert current sexp into multi-line	M		Extend current sexp into multiple lines. Especially useful on results of macroexpand.
Turn current sexp into one line	O		Turn current sexp into one line.
Stringify current sexp	S		Transform current sexp into a string. Quote newlines if arg isn't 1.
Insert a space	Space		
Outline operations			
Toggles on off org-mode-like outline	I		Toggles on/off an org-mode-like outline. <ul style="list-style-type: none"> To make this work, lispy-mode will modify outline-regexp and outline-level-function for the current buffer while it's on.
Indent / hide/show outline	i	<ul style="list-style-type: none"> With no active region: (special-lispy-tab) 	If in outline: hide/show outline, otherwise indent all code of current paren <ul style="list-style-type: none"> When region is active, call 'lispy-mark-car'.
Next outline level	J		Takes a numeric prefix arg and calls outline-next-visible-heading arg times or until past the last outline-regexp.
Previous outline level	K		Takes a numeric prefix arg and calls outline-previous-visible-heading arg times or until past the first outline-regexp.
Evaluate Code			
Eval last sexp	e	(special-lispy-eval ARG)	Eval last sexp. Display result in echo area. <ul style="list-style-type: none"> When ARG is 2, insert the result as a comment.
Eval current region sexp. Insert result.	E		Eval current region or sexp. The result will be inserted in the current buffer after the evaluated expression.
Eval current sext & replace it at point	x r		
Eval current sexp in the content of the of the other window	p		
EDebug current defun	x e		edebug current defun. Or cider-debug-defun-at-point for Clojure.
	2x e		2xe will eval current defun instead.
Debug - step in	x j		<ul style="list-style-type: none"> Evaluate the arguments at the current function's call Jump to the function's definition Set the result of evaluation to the function's arguments
EDebug stop	z		Does the same as q in edebug, except current function's arguments will be saved to their current values. <ul style="list-style-type: none"> This allows to continue debugging with lispy-eval (e) from edebug's current context. The advantage is that you can edit the code as you debug, as edebug puts your code in read-only mode.
Execute Tests: run ert	x T		
Buffer/Region operations			
Store current buffer and region for further operation	x B		
Ediff regions	B		