Lispy minor mode

<u>Description</u>	Key	Function	<u>Note</u>			
<u>Lispy</u> - Single letter commands to manipulate Lisp- like code	The lispy mode provides modal editing to Emacs for Lisp-like languages. Lisp is a very structured programming language, made of succession and combinations of S-expressions ("sexp"): lists that start with (and end with) "paren". As long as point (the cursor) is before the left, opening, paren or the right, closing paren, the keys are interpreted as lispy commands. Keys in other locations are interpreted as usual. The nature of the Lips programming languages enables this type of modal editing that is even more powerful than Vi-type modal editing.					
		This table lists the lispy command keys, with links to the <u>Lispy function Reference</u> for each one.				
T Out to the DEL and	This requires the <u>lispy</u> extends the <u>lispy</u>	rernal package. PEL do (pel-cfg-pkg-lisp	ownloads, installs and activates lispy when the pel-use-lispy user option is set to t. Customize support for Lisp programming languages - A group that also contains the groups for			
∑ Customize PEL and Emacs Lispy support	CIII> CIZ> SPC M-L	&optional OTHER- WINDOW)	Emacs Lisp and Common Lisp: lispy. If OTHER-WINDOW is non-nil (use C-u), display in another window.			
Getting Help on code						
Describe function at point	• C-1 • <f12> 1</f12>	(lispy-describe-inline)	Display documentation for 'lispycurrent-function' inline. • If docstring is small enough it is displayed in a pop-up box above point. Otherwise it is			
See Also: <u>∑ Help/Info</u>			displayed inside a *lispy-help* buffer. This requires the <u>lispy</u> external package. DEL downloads, installs and activates lispy when the pel-use-lispy user option is set to t.			
Describe function arguments	• C-2 • <f12> 2</f12>	(lispy-arglist-inline)	Show the argument list of current function.			
Describe function/ variable	xh		A shorthand for describe-function or describe-variable. If you want to call describe-variable, you should mark the symbol first.			
Navigate inside code	The following commands move point inside code when point is before left paren or after right paren.					
ace symbol move	a	(special-lispy-ace- symbol ARG)	Jump to a symbol within the current sexp and mark it. Use ace method: each symbol in sexp is shown with highlight letter: type that letter to move to the symbol. Sexp is obtained by exiting the list ARG times.			
ace sub-word	-		Similar to lispy-ace-symbol, but selects a subword instead.			
Move back	b	(special-lispy-back ARG)	Move point to ARGth previous position in lisps-back history If position isn't special, move to previous or error. Lispy back history updated by: 1, h, f, j, k, m, q, and i.			
Move to different (other) side of sexp	đ	(special-lispy- different)	Switch to the different side of current sexp. • If before '(' move after ')' and vice-versa.			
Flow: move in the direction of current paren	f	(special-lispy-flow ARG)	Move in the direction of current paren: At left: move to next left paren (move going down the file). At right: move to previous right parent (move going up the file). Don't enter strings or comments.			
Move left outward	h	(special-lispy-left ARG)	Move outside list backwards ARG times. Return nil on failure, t otherwise.			
Move down current list	j	(special-lispy-down ARG)	Move down ARG times inside current list. • Guaranteed to never exit the list: 99j moves to the last element of the current list. • Updates lispy-back history.			
Move up current list	k	(special-lispy-up ARG)	Move up ARG times inside current list. • Guaranteed to never exit the list: 99k moves to the first element of the current list. • Updates lispy-back history.			
Start knight hydra	z		Start/Terminate the knight hydra			
Move down left-most parens on each line	• zj • j		Move down left-most paren to the next line (can exit list)			
Move up left-most parens on each line	• zk • k		Move up left-most paren to the previous line (can exit list)			
Move outside list forward	1	(special-lispy-right ARG)	Move outside list forwards ARG times. Parens in strings and comments are ignored. Updates lispy-back history.			
Move to Ace target	đ		Highlights each symbol in current sexp as ace target and jump to the selected one. • Updates lispy-back history.			
Move to Ace target char	Q		Prompts for character, highlights each one in current sexp as ace target and jump to the selected one.			
View: center current sexp	v		Recenter current sexp to be on the first line of the window. When called twice in a row, recenter back to the original position.			
Visit another file	v		Visit another file within this project using projectile or find-file-in-project (customize lispy-visit-method to choose). • Use V to call projectile-find-file. Use 2V to call projectile-find-file-other-window.			
Search						
Occur search inside the current top-level sexp	У		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 using directory tabgs	g	(special-lispy-goto &optional ARG)	Jump to symbol within files in current directory. Prompt for symbol and jump to it. • When ARG isn't nil, call 'lispy-goto-projectile' instead. • See lispy goto wiki page.			
goto definition using projectile base directory	• 0g • ogp	(lispy-goto-projectile)	Jump to symbol within files in ('projectile-project-root').			
goto definition in local file	G		Similar to lispy-goto, but only current file's tags are used instead of whole directory's tags.			
Follow: jump to definition	• F • M		When region is active jump to the definition of marked symbol. Otherwise jump to the definition of the first symbol in current sexp.			
Pop tag	• D • M-,		Go back from where it came with Follow			
Narrow/Widening See also: Narrowing	Narrowing hides everythin Widen it back to see the c		selected region, allowing work on that region alone.			

previous sexp as replacement for their parent. Move sexp down in list s Move current sexp or region down arg times. Don't exit the parent list. Also works for outline undo u Move current sexp up Move current sexp up Move current sexp to region up arg times. Don't exit the parent list. Also works for outlines. Transform the current list expression into a let-bound variable, ledit-mode is used to name to new variable. Use M-m to finish naming the variable, ledit-mode is used to name to new variable. Use M-m to finish naming the variable. Unbind a let bound variable at turn current lambda into a defun turn current defun into a lat ledit into cond turn cond into nested if expressions Furn current defun into a lat ledit into cond turn cond into nested if expressions Furn current turn current turn cond into nested if expressions Furn current turn current lambda into a call late and late and late body find-able. Convolute: Exchange the order of application of 2 closes to uter forms, relative to current expression region. Surp: grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beavan or region. Barf: shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beavan or region. Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beavan or region. Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beavan or region. Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beavan or region. Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beavan or region. Shrink either current list into the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction.	<u>Description</u>	Key	Function	Note
Court Facility Court C		N		Narrow current sexp or region.
With active region	Widen	w		Widen back to see the complete buffer.
mark car if with society region is active, call lapp-mark-car? Copy region a save 16 in paceuti slapp-mark 10 in paceu				
Siley-mark-card Siley-mark-card Siley-mark history Copy region or seap to bill ring. Siley-mark history Siley-mark Siley-mark history Siley-mark		i	•	· · · · · · · · · · · · · · · · · · ·
Mark list	mark car	i		
Paste Paste Paste Income the Income And income the Income And income the Income And Inco		n		Copy marked region or sexp to kill ring.
Edit code close cl	Mark list	m		When ARG is more than 1, mark ARGth element.
clone clone clone clone clone clone clone (special-liapy-clone ARG times. 'When the surp is top level, insert an additional newline. clone surprise sept to the left control of the current sept to the left clone current sept to region down any limes. Don't exit the parent list. Also works for culting clone current sept to the parent list. Also works for culting clone current sept to the parent list. Also works for culting clone current sept to the parent list. Also works for culting clone current sept to the parent list. Also works for culting clone current sept to the parent list. Also works for culting clone current sept to the parent list. Also works for culting clone current sept to the parent list. Also clone clone current sept to the current sept to the current sept to the parent list. The current sept t	Paste	P		When region is active, replace it with current kill. Forward to yank otherwise.
ARGS When the suxp is top loved, insent an additional newline. When the suxp is top loved, insent an additional newline. When the suxp is top loved, insent an additional newline. When current sexp inside first element of list below Move current sexp to the first element of list below. Move current sexp to the first during current sexp as a replacement for list above. Where current sexp to the first the large and the following (if called from the left), or the preceding (if called from the left),	Edit code	Transform code using the fo	lowing commands	
More current sexp inside first element of list below More current sexp to the first claims current sexp are replacement of list below More current sexp to the first claims current sexp are replacement of list below More current sexp are replacement of list list above I list above	<u>clone</u>	С		
first element of list below More current says to be become list element of list above More current says to be list above More current says to be list above More current says to be list above I l		oh		
Become last element of list above Move current sexp to the rich Loudside current sexp and sexp and the following (if called from the left), or the percending (if called from the percending (if called from the left), or the percending (if called		oj		
Fig. 1: Land transfer current list Move current sexp as replacement for this parent sexp as replacement for their parent. Rises: current and noxt previous sexp as replacement for their parent. Rises: current and noxt previous sexp as replacement for their parent. Rises: current and noxt previous sexp as replacement for their parent. Rises: current sexp to their parent. Rises: current sexp to their parent. Rows sexp down in list s s Move current sexp or region down arg times. Don't exit the parent list. Also works for outline undo u Move current sexp to the current sexp to the bound variable. Rised var: current sexp to the bound variable to the list hand variable to the list hand variable. Rised var: current sexp to the bound variable to the list hand variable. Rised var: current sexp to the bound variable hand to the list hand variable to the list hand va	become last element of	ok		
Basecurent and next previous sexp as replacement for the previous sexp as replacement for their parent. Basecurent and next previous sexp as replacement for their parent. Base current sexp or the active region as replacement for their parent. Base sexp down in list a Move current sexp or region down and times. Don't exit the parent list. Also works for outline undo u Move current sexp por which are the parent list. Also works for outline. Bind varr. current sexp to be a more than the current list expression into a let-bound variable; ledit-mode is used to name the bound variable. Unbind a let bound variable undo a defan a sexp and the following if called from the first hard to finish naming the variable; ledit-mode is used to name the bound variable. Unbind a let bound variable undo a defan a sexp and the following and the current list expression into a let-bound variable; ledit-mode is used to name the bound variable. Unbind a let bound variable undo a defan a sexp and the following into a let-bound variable. Also works for Clojure. **Transform the current list expression into a let-bound variable. Also works for Clojure. **Transform the current list expression into a let-bound variable. Also works for Clojure. **Transform the current flat into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current affect into a let-bound variable. Also works for Clojure. **Into current flat into a let-bound variable. Also works for Clojure. **Into current flat into a let-bound variable. Also works for Clojure. **Into current flat into a let-bound variable. Also works fo		ol		
previous sexp as replacement for their parent. replacement for their parent. replacement for their parent. replacement for their parent. Move sexp down in list s Move current sexp or region down arg times. Don't exit the parent list. Also works for outline. Move current sexp up W	as replacement for its	r		
Move current sexp up W Move current sexp or region up arg times. Don't exit the parent list. Also works for outlines. Bind var. current sexp to xb Transform the current list expression into a let-bound variable; iedit-mode is used to name to the bound variable. W W W W W W W W W	previous sexp as replacement for their	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.
Move current sexp.up Move current sexp to the thought surple to the common to the com	Move sexp down in list	s		Move current sexp or region down arg times. Don't exit the parent list. Also works for outlines.
Transform the current list expression into a let-bound variable; iedit-mode is used to name to new variable. Use M-m to finish naming the variable. Vumbind a let bound variable. Use M-m to finish naming the variable. Vumbind a let bound variable. Also works for Clojure. Vum current lambda into a defun variable. Also works for Clojure. Vum current defun into a lambda vum current defun into a lambda vum current defun into a valuant of the variable. Vum current defun into a lambda vum current function or variable. Vum cond into nested if vai expressions vum current function or variable. Vum cond into nested if vai expressions vum current function or variable. Vum cond into nested if vai expressions vum current function or variable. Vum cond into nested if vai expressions vum current function or variable. Vum current expression region. Carow either current expression region. Carow either current expression. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow variable variable. Vum current expression. Shirink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-to-variable variable into possible before that position. Shirink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-to-variable variable. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of 0, grow as far as possible. Vum an arg of	<u>undo</u>	u		
Inline al et bound variable	Move current sexp up	w		Move current sexp or region up arg times. Don't exit the parent list. Also works for outlines.
variable xd turn current defun into a defun xd turn current defun into a lambda x1 turn nested if into cond xc turn cond into nested if expressions xi linine current function or macro call. xi Convolute: Exchange the order of application of 2 closest outer forms. c Convolute: Exchange the order of application of 2 closest outer forms. c Sturce arow either current sexp or region. Sex possible before that position. Sturce arow either current sexp or region of the possible sexp or region. Splice the current sexp or region (if it's active) in appropriate direction. Opposite of lispy-beside or sexp or region. Splice the current list into the parent list into the parent list into the parent 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 the point to the parent list in appropriate direct forward to beginning-of-defun. When called twice in a row, restore the previous point and no positions. Teleport: move current sexp to Ace target symbol. A crass to replace Move current sexp to Ace target inside current function 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.		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.
turn current defun into a lambda lambda lambda lambda lambda lurn nested if into cond xc turn cond into nested if into cond xc turn cond into nested if expressions 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. relative to current expression region. Convolute: Exchange the order of application of two closest outer forms, relative to current expression region. Slurp: grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-besive or region. Barf: shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-besivo or region. Barf: shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-besivo or region. 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. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the		xu		Unbind a let-bound variable. Also works for Clojure.
turn cond into nested if into cond turn cond into nested if turn cond into nested if expressions 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 Surpe grow either current expression region. Crow either current sexp or region if it's active) in appropriate direction. Opposite of lispy-bear of the function of the current sexp ends or as far as possible. 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. 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. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list in appropriate direction. If there are none within the parent list in appropriate direction. If there are none within the parent list in appropriate direction. If there are none within the parent list in a province the previous point and no positions. Teleport: move current sexp to Ace target symbol are also to the parent list in appropriate direction. Open and the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list in a province the parent lis		xd		
turn cond into nested if expressions Inline current function or macro call intermed and its body find-able. Convolute: Exchange the order of application of 2 closest outer forms. Slurp: grow_either.current sexp or region. Slurp: grow_either.current sexp or region (if it's active) in appropriate direction. Opposite of lispy-bears or region. Slurp: grow_either.current sexp or region (if it's active) in appropriate direction. Opposite of lispy-bears or region. Shirp: grow_either.current sexp or region (if it's active) in appropriate direction. Opposite of lispy-bears or region. Shirp: grow_either.current sexp or region (if it's active) in appropriate direction. Opposite of lispy-bears or region. Shirp: grow_either.current sexp or region (if it's active) in appropriate direction. Opposite of lispy-bears or region. Shirp: grow_either.current sexp or region (if it's active) in appropriate direction. Opposite of lispy-bears or region. 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 direct the parent list of beginning-of-defun. When called twice in a row, restore the previous point and in current defun. Teleport: move current sexp to Ace target inside current function sexp to Ace target to any sexp inside current window. Move to Ace target to any sexp inside current window. Move to Ace target to any sexp inside current window. Convert current sexp into multi-line. Minimal current sexp into multi-line. Turn current sexp into one line.		x1		
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. Slurp: grow either current sexp or region. Slurp: grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-be with an arg of 0, grow as far as possible. 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 possible before that position. Barf: shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-be possible before that position. 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. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list 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. Teleport: move current sexp to Ace target inside current function sexp to Ace target the Move current sexp to Ace target to any sexp inside current window. Move to Ace target which is a proposed and believe the selected symbol. Turn current sexp into multiple lines. Especially useful on results of macroexpand. Turn current sexp into one line.	turn nested if into cond	жс		
Interned and its body find-able. Convolute: Exchange the order of application of 2 closest outer forms		xi		
region. Surp: grow either current sexp or region		xf		Inline current function or macro call, i.e. replace it with function body. The function should be interned and its body find-able.
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 (if it's active) in appropriate direction. Opposite of lispy-sexp or region 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. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list, move to the parent list in appropriate direction. If there are none within the parent list. Move the parent list in appropriate direction. If there are none within the parent list. Move the parent list in appropriate direction. If there are none within the parent list. Move the parent list into the parent list. Move the parent list. Move the parent list into the parent list. Move the parent list. Move the parent list. Move the parent list into the parent list. Move the parent list. Move the parent list. Move the parent list. Move the parent list into the parent list. Move the parent list. Move the parent list. Move the parent list	order of application of 2	С		Exchange the order of application of two closest outer forms, relative to current expression or region.
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 direct. Move to beginning of current defun A Forward to beginning-of-defun. When called twice in a row, restore the previous point and in positions. Teleport: move current sexp to Ace target inside current function **Exemplate** **Exemplate** **Exemplate** **Extend current sexp into one line.** **Turn current sexp into one line.** **Turn current sexp into one line.** **Splice the current list into the parent list. Move the point to the next list to splice in appropriate direct in appropriate direct. **Splice the current list into the parent list. Move the parent list. Move to the parent list in appropriate direct. **A Power of the parent list. Move the point to the next list to splice in appropriate direct. **A Move to Ace target in a row, restore the previous point and in positions. **Courrent sexp to Ace target inside current function **Extend current sexp into any sexp inside current window.** **Calls lispy-ace-symbol and deletes the selected symbol.** **Extend current sexp into multiple lines. Especially useful on results of macroexpand.** **Turn current sexp into one line.** **Turn current sexp into one line.**		>		• With an arg of -1, grow until the end of the line where the current sexp ends or as far as
the parent list Move to beginning of current defun Teleport: move current sexp to Ace target symbol & erase to replace Convert current sexp into multi-line Turn current sexp into one line Move to beginning of current list, move to the parent list, move to the parent list in appropriate direct direction. If there are none within the parent list, move to the parent list, move to the parent list in appropriate direct A provard to beginning-of-defun. When called twice in a row, restore the previous point and no positions. Move current sexp to Ace target inside current function Move to Ace target to any sexp inside current window Calls lispy-ace-symbol and deletes the selected symbol. Extend current sexp into multiple lines. Especially useful on results of macroexpand. Turn current sexp into one line.		<		Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-slurp.
current defun positions. Teleport: move current sexp to Ace target t Move current sexp to Ace target inside current function tet 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 Turn current sexp into one line.		/		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.
Teleport: move current sexp to Ace target t		A		
Move to Ace target sexp into Ace target to any sexp inside current window Move to Ace target symbol & erase to replace 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.		t		
symbol & erase to replace Convert current sexp into multi-line Extend current sexp into multiple lines. Especially useful on results of macroexpand. Turn current sexp into one line. Turn current sexp into one line.	Sexp to Ace target	tt		Move current sexp to Ace target to any sexp inside current window
multi-line Turn current sexp into one line. Turn current sexp into one line. Turn current sexp into one line.	symbol & erase to	н		Calls lispy-ace-symbol and deletes the selected symbol.
one line		м		Extend current sexp into multiple lines. Especially useful on results of macroexpand.
Stringify current sexp s Transform current sexp into a string. Quote newlines if arg isn't 1.		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	Insert a space	Space		
Outline operations	Outline operations			
Toggles on/off org-mode-like outline. Toggles on/off an org-mode-like outline. To make this work, lispy-mode will modify outline-regexp and outline-level-function for the current buffer while it's on.		ı		To make this work, lispy-mode will modify outline-regexp and outline-level-function for the

Description	Key	Function	<u>Note</u>
Indent / hide/show outline	i	With no active region: (special-lispy-tab)	If in outline: hide/show outline, otherwise indent all code of current paren • 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	К		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	е	(special-lispy-eval ARG)	Eval last sexp. Display result in echo area. • When ARG is 2, insert the result as a comment.
Eval current region sexp. Insert result.	Е		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	xr		
Eval current sexp in the content of the of the other window	p		
EDebug current defun	xe		edebug current defun. Or cider-debug-defun-at-point for Clojure.
	2xe		2xe will eval current defun instead.
<u>Debug - step in</u>	×j		 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. 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	хT		
Buffer/Region operations			
Store current buffer and region for further operation	хВ		
Ediff regions	В		