

Description	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.  This requires the <u>lispy</u> external package. PEL downloads, installs and activates lispy when the <u>pel-use-lispy</u> user option is set to t.				
∑ Customize PEL and Emacs Lispy support	<f11> <f2> SPC M-L</f2></f11>	(pel-cfg-pkg-lisp &optional OTHER- WINDOW)	Customize support for Lisp programming languages - A group that also contains the groups for Emacs Lisp and Common Lisp: lispy.  • If OTHER-WINDOW is non-nil (use <b>C-u</b> ), display in another window.		
Getting Help on code		Wilder	TOTALITE WINDOW IS HOFFIIII (USE C-W), display in another window.		
Describe function at point  See Also:   Help/Info	• 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 displayed inside a 'lispy-help' buffer.  This requires the <u>lispy</u> external package. PEL 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 mo	ove point inside code when	n 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	d	(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.  • Updates lispy-back history.		
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	( <b>special-lispy-right</b> ARG)	Move outside list forwards ARG times.  Parens in strings and comments are ignored.  Updates lispy-back history.		
Move to Ace target	ď		Highlights each <b>symbol</b> 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 <b>in current directory</b> . Prompt for symbol and jump to it.  • When ARG isn't nil, call 'lispy-goto-projectile' instead.  • See <u>lispy goto wiki page</u> .		
goto definition using projectile base directory	• 0g	(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: <u>∑ Narrowing</u>	Narrowing hides everythin     Widen it back to see the c		selected region, allowing work on that region alone.		

With an arg of 0, grow as far as possible.	<u>Description</u>	Key	Function	<u>Note</u>
Court Person Mariety  Interest Control		N		Narrow current sexp or region.
Mile	Widen	W		Widen back to see the complete buffer.
mark.car				
Segmentation of the content of the		i		· • • • • • • • • • • • • • • • • • • •
Second	mark car	i		
March ARGS   more than 1, mak ARGS   more than 1, ma		n		Copy marked region or sexp to kill ring.
Edit code    Comparison Code using the Distriction Commands   Comparison Code using the Distriction of April Imperiodic Applications (Applications)   Comparison Code using the Distriction of Applications (Applications)   Comparison Code using the Distriction Code	Mark list	m		When ARG is more than 1, mark ARGth element.
Cobbe Search 1	<u>Paste</u>			When region is active, replace it with current kill. Forward to yank otherwise.
ARC)  When the seep is top level, insert an additional newline.  When the seep is top level, insert an additional newline.  When current seep and the following it called from the lattic of the become last demand of last below.  Move current seep and the following it called from the lattic, or the proceeding it called from the lattic, or the proceeding it called from the previous seep as explacement for their parent.  Raise: current seep and the following it called from the lattic, or the proceeding it called from the previous seep as explacement for their parent.  Raise: current seep and the following it called from the lattic, or the proceeding it called from the previous seep as explacement for their parent.  Raise: current seep and the following it called from the lattic, or the proceeding it called from the previous seep as explacement for their parent.  Raise: current seep and the following it called from the lattic, or the proceeding it called from the previous seep as explacement for their parent.  Raise: current seep and the following it called from the previous seep as explacement for their parent.  Raise: current seep and the following it called from the previous seep as explacement for their parent.  Raise: current seep to make the parent seep to region down any times. Dort exit the parent see, how one of the parent seep to region down any times. Dort exit the parent see, how one of the parent seep to region the seed of the parent seep to the parent seep to region in the court of the parent seep to region in the court seed the parent seep to region if it is actively in appropriate direction. Opposite of lapsy bard seed to the parent seep to region if it is actively in appropriate direction. Opposite of lapsy bard seep to region of it is actively in appropriate direction. Opposite of lapsy bard seep to region of it is actively in appropriate direction. Opposite of lapsy bard seed to the pa	Edit code	Transform code using the fol	lowing commands	
More current sexp inside first all seminant of list below More current sexp to become last derivent of list below More current sexp to become last derivent of list above More current sexp to become last derivent of list above More current sexp to be second last derivent of list above More current sexp to the last derivent of list above More current sexp to the last derivent of list above  All current sexp to the last derivent of list above  All current sexp and the foliaving of called from the left), or the preceding of called from the right of sexp or region down and primes. Don't exit the parent sex Also works for outlines.  Move scurrent sexp to make the parent sex Also works for outlines.  Move scurrent sexp to make the parent sex Also works for outlines.  Move scurrent sexp to make the parent sex Also works for outlines.  Move scurrent sexp to make the parent sex Also works for outlines.  Blind star. current associate  Listend as the bound standard  to the last bound standard in the current standard list of the last bound standard  to the last bound standard in the current standard list of the last bound standard in the current standard list of the last bound standard in the current standard list of the last bound standard in the current standard list of the last bound standard in the current standard in the cur	<u>clone</u>	С		
Interestment of list below   Whose current seaso to be become hast element of list above   Second has element   Second hast element of list above   Second has element		oh		
Decome last element of last above with the state above with the state above with the state of th		oj		
Find the control last Relieuses current seap as replacement for the parent seap are replacement for their parent.  Relieuse current and next providus sexps as replacement for their parent.  Relieuse current and next providus sexps as replacement for their parent.  Relieuse current and next providus sexps as replacement for their parent.  Relieuse pown in list used to sexps and the following if called from the proceeding (if called from the providus sexps as replacement for their parent.)  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down in list used to sexps as replacement for their parent.  Rove sexps down arg times. Don't exit the parent list. Also works for outlines.  Rove sexps are replacement as alter the parent list. Also works for outlines.  Rove sexps are replacement as alter the parent list. Also works for colique.  Rove sexps are replacement as alter than a sexps are replacement as alter than a sexps are replacement.  Rove sexps are replacement as alter than a sexps are replacement.  Rove sexps are replacement as alter than a sexps are replacement.  Rove sexps are replacement as alter than a sexps are replacement.  Rove sexps are replacement as alter than a sexps are replacement.  Rove sexps are replacement as alter than a sexps are replacement.  Rove sexps are replacement as alter than a sexps are replacement as altered than a sexps are replacement.  Rove sexps are replacement as altered than a sexps are replacement as altered to a sexps are replacement.  Rove sexps are replacement as altered than a sexps are replacement as altered as a s	become last element of	ok		
Raises current and next previous sexps as replacement for their parent.  Raises current and next previous sexps as replacement for their parent.  Raises current as sexp and the following if called from the left), or the preceding (if called from the right) sexps, or the active region as replacement for their parent.  Move sexp down hill it is Move current sexp over region down arg times. Don't exit the parent list. Also works for outlines. Move current sexp up  Move current sexp up  Move current sexp up  Billed var, current sexp to the bound variable to bound variable to the more than the parent list. Also works for outlines.  Unbind a let bound variable and a set to the more than the parent list. Also works for Clojure.  Variable  Unbind a let bound variable. Also works for Clojure.  Variable  Variab		ol		
right) sexpo, or the active region as replacement for their parent.  right) sexpo, or the active region as replacement for their parent.  Move sexp down in list sundo  whose sexp down in	as replacement for its	r		
Move current sexp. up  Bind var: current sexp. up  to be thound variable  Lobind a let bound  variable  Unbind a let bound  variable  Unbind a let bound  variable. Use M-m to finish marning the variable.  Lob war variable. Use M-m to finish marning the variable.  Variable  Variable	previous sexp as replacement for their	R		
Move current sexp up Bind var. current sexp to the technology of t	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; ledit-mode is used to name the new variable. Use M-m to finish naming the variable. With the current list expression into a let-bound variable; ledit-mode is used to name the new variable. Use M-m to finish naming the variable. With the part of the late of the part of the pa	<u>undo</u>	u		
Institute of the current search of the cur	Move current sexp up	w		Move current sexp or region up arg times. Don't exit the parent list. Also works for outlines.
variable         variable         xd           turn current lambda into a defun         xd         xd           turn unrent defun into a lambda         x1         x1           turn cond into nested if embda         x1         x1           turn cond into nested if expressions         x1         x1           linine current function or macro call it.e. replace it with function body. The function should be interned and its body find-able.         x1           Convolute: Exchange the order of application of 2 application of 3 application of 2		xb		
turn current defun into a lambda  turn nested if into cond  turn cond into nested if sit cond   xc    turn cond into nested if sit cond   xc    turn cond into nested if sit   xi    turn cond into nested into nested into nesults of macroexpand.  Turn current sexp into one line.  Turn current sexp into a string. Quote newlines if arg isn't 1.		xu		Unbind a let-bound variable. Also works for Clojure.
lambda         xc           turn nested if into cond         xc           turn cond into nested if expressions         xi           sypressions         xf         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.           Slure; grow either current sexp or region         c         Grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-burf-sexp or region           Barf: shrink either current sexp or region         Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-burf-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 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 a		xd		
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  Convolute: Exchange the order of application of 2 closest outer forms  Slurp: grow either current expression or region.  Splice the current list in expropriate direction. Opposite of lispy-slur expression or region.  Splice the current list. Move the point to the next list to splice in appropriate direction. Opposite of lispy-slur expression. If there are none within the parent list, move to the parent list in appropriate direction. Opposite of grow expression.  Move to beginning of a proving of grow expression or region. If there are none within the parent list, move to the parent list in appropriate direction. Opposite of grow expression. If there are none within the parent list, move to the parent list in appropriate direction. Opposite of grow expression. If there are none within the parent list, move to the parent list. Move the parent list. Move the point to the next list to splice in a		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  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-barf.  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 or region (if it's active) in appropriate direction. Opposite of lispy-barf.  With an arg of 1, grow until the end of the line where the current sexp ends or as far as possible.  Barf: shrink either current  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 parent list, move to the parent list in appropriate direction. If there are none within the par	turn nested if into cond	xc		
Interned and its body find-able.		xi		
region.  Grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-barf.  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-slur sexp or region  Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur 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. Move to beginning of current defun  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 inside current function  Move to Ace target symbol.  Move current sexp to Ace target to any sexp inside current window  Calls lispy-ace-symbol and deletes the selected symbol.  Symbol & erase to replace  Convert current sexp into multiple lines. Especially useful on results of macroexpand.  Turn current sexp into one line.  Turn current sexp into a string. Quote newlines if arg isn't 1.  Insert a space  Space  Outline operations		xf		
Sturp: grow either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-barf.  • 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.  Shrink either current sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or possible.  Splice the current sexp or region (if it's active) in appropriate direction. Opposite of lispy-slur sexp or appropriate direction. Opposite of lispy-slur sexp or appropriate direction. Opposite of lispy-slur appropriate direction. Opposite of lis	order of application of 2	С		
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 inside current function  Move to Ace target  tt Move current sexp to Ace target to any sexp inside current window  Calls lispy-ace-symbol and deletes the selected symbol.  Convert current sexp into multiple lines. Especially useful on results of macroexpand.  Turn current sexp into one line.  Stringify current sexp  Space  Outline operations		>		• With an arg of -1, grow until the end of the line where the current sexp ends or as far as
the parent list  direction. If there are none within the parent list, move to the parent list in appropriate direction  Move to beginning of current defun  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  the Move current sexp to Ace target inside current function  Move to Ace target  Move to Ace target symbol & erase to replace  Convert current sexp into multi-line  Turn current sexp into one line  Stringify current sexp  Space  direction. If there are none within the parent list, move to the parent list in appropriate direction  Forward to beginning-of-defun. When called twice in a row, restore the previous point and mark positions.  Move current sexp to Ace target inside current function  Move to Ace target 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.  Turn current sexp into one line.  Stringify current sexp  Space  Outline operations		<		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       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     0     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		/		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 current sexp to Ace target inside current function		A		
Move to Ace target sexp inside current window  Move to Ace target symbol & erase to replace  Convert current sexp into multiple lines. Especially useful on results of macroexpand.  Turn current sexp into one line.  Turn current sexp into one line.  Stringify current sexp  Space  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.  Turn current sexp into one line.  Transform current sexp into a string. Quote newlines if arg isn't 1.  Insert a space  Outline operations		t		
Symbol & erase to replace  Convert current sexp into multiple lines. Especially useful on results of macroexpand.  Turn current sexp into one line.  Turn current sexp into one line.  Stringify current sexp  Space  Transform current sexp into a string. Quote newlines if arg isn't 1.  Insert a space  Space  Outline operations	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.       Stringify current sexp     S       Insert a space     Space       Outline operations	symbol & erase to	н		Calls lispy-ace-symbol and deletes the selected symbol.
One line     Stringify current sexp     Stringify current sexp into a string. Quote newlines if arg isn't 1.       Insert a space     Space       Outline operations		М		Extend current sexp into multiple lines. Especially useful on results of macroexpand.
Insert a space Space Outline operations		0		Turn current sexp into one line.
Outline operations	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 an 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.	Toggles on off org-mode- like outline	I		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.
<b>Evaluate Code</b>			
Eval last sexp	е	( <b>special-lispy-eval</b> 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>	xj		<ul> <li>Evaluate the arguments at the current function's call</li> <li>Jump to the function's definition</li> <li>Set the result of evaluation to the function's arguments</li> </ul>
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 readonly mode.
Execute Tests: run ert	хT		
Buffer/Region operations			
Store current buffer and region for further operation	жВ		
Ediff regions	В		