## **PEL Key Maps**

PEL Ney Maps						
<u>Operation</u>	<u>Keystroke</u>	Key Map	Note			
Emacs Key Bindings	Emacs has a large set of key b		ally in order the letter (a) in the assumpt by ffer			
See also: <u>Name Modifier Keys</u>	<ul> <li>Some commands are bound to single keys like the a key which normally inserts the letter 'a' in the current buffer.</li> <li>Some commands are bound to functions keys like <f1> or use key modifiers like C-a or M-a. See Modifier Keys for more info.</f1></li> <li>Some commands are bound to longer key sequences lie C-x s.</li> <li>The first key, or the first set of keys, can be used as an Emacs key prefix. And then several other keys can follow, all under that prefix. The prefixereates some sort of scope: the key-map under that prefix.</li> </ul>					
	<ul> <li>There's really no limit to the way you can combine keys, the modifier keys, with or without short or longer key prefixes.</li> <li>On top of that you can have key bindings that are</li> <li>global, always accessible if the related code was loaded, or</li> <li>local, only available while a specific major or minor mode is activated inside a specific buffer.</li> <li>All of this provides great flexibility. But it makes Emacs more difficult to learn: you need to remember all the keys.</li> </ul>					
PEL Key maps	Although PEL itself adds a large amount of keys to what's already in Emacs, it leaves most Emacs key binding intact and mainly uses the function keys organized under a tree of key prefixes, trying to provide easy-to-remember key prefixes.					
See also: <u><b>Keys - Fn</b></u>	<ul> <li>PEL key bindings are accessible from Emacs running in graphics mode and in terminal mode (you may have to configure your termcap terminal software to support ASNI key sequences for function and cursor keys).</li> <li>By default, PEL also activates the which-key external package which allows you to see all command key bindings for each key prefix in the echo area at the bottom of your Emacs screen.</li> <li>PEL provides documentation of the Emacs and PEL key bindings, organized in topics inside PEL files such as this one.</li> <li>All PEL key prefix groups provide a <f1> key binding to a command that opens a local copy of a PDF file describing the topic. To open this PDF file from Emacs using PEL, just type <f1> <f1> The <f11> key is the most often used PEL global key prefix. Inside its group the <f1> key opens this file.</f1></f11></f1></f1></f1></li> </ul>					
	This page lists PEL's key maps.  Column 1, the title column, shows the name of the PEL specific PDF page and it's also a link to the Github hosted pdf page.  Column 2 shows the key sequence for the topic.  Column 3 shows the name of PEL key prefix for the topic.					
	Some topics do not have common PDF tables. These are listed file	nmands organized under on specific PEL key map, but the commands and keys are described inside topic specific first set of rows below.				
	Firefox will open the PDF files and will render it inside the browser page instead of downloading it.  This is a great way to navigate through the various links if you are online. For other browsers, you may have to install pdf rendering plugins to do the same.					
Topics with no PEL key maps	The following topics do not have a PEL topic-specific key-map.  You can use the <f11> ? p key sequence and enter the topic name to open the file. The command support tab completion. See <u>Nelp/Info</u></f11>					
<u>≻Legend</u>	Describes all conventions and symbols used in the PEL PDF files.					
M AsciiDoc	AsciiDoc support					
∑ Autosave/Backup	Emacs commands for autosave and backup control					
∑ Case Conversions	Commands for case conversion of text.					
∑ Closing/Suspending	Commands to close or suspend Emacs.					
∑ Completion/Input	Commands to complete user input at prompts.					
<u>∑M CUA</u>	CUA mode commands.					
∑ Enriched Text	Commands that support the enriched text concept.					
<u>¥ ERT</u>	Emacs Lisp unit testing commands.					
∑ Faces/Fonts	Commands that control Emacs faces and fonts.					
<u> </u>	Commands to enable/disable key chords (typing 2 normal keys together to invoke a command).					
<u>■Keys - Fn</u>	Table that shows the way PEL uses function keys.					
M Outline/Org-Mode	Org-mode commands.					
<u> </u>	Describes Emacs modifier keys and ways of describing keys in Emacs.					
<u>∑ Mouse</u>	Mouse commands. Available both in graphics and terminal modes.					
<u>Narrowing</u>	Narrowing commands. A way to narrow your view to only a portion of the current buffer, protecting the rest of the buffer from any modification.					
<b>∑</b> Navigation	The navigation commands available in Emacs with the additions provided by PEL and other packages.					
<u>∑</u> Numkeypad	Describes the way the numerical keypad is handled in Emacs.					
<u></u> Packages	Commands to download and manipulate external packages.					
<u>∑ Rectangles</u>	Commands to manipulate recta	angle areas of text inside a buffer.				
∑ Semantic	Planned topic					
∑ SyntaxCheck	Planned topic					
Global Key Maps	The key maps are listed in order	The key maps are listed in order of the key they use. The keys were selected mnemonic naming as much as possible. For that reason some key maps are accessible via several key prefix sequences.				
Top level prefix	<f11></f11>	pel:	Key prefix			
<u>∑ Indentation</u>	<f11> TAB</f11>	pel:indent				
∑ Spell Checking	<f11> \$</f11>	pel:spell				
<u></u> Bookmarks	<f11> '</f11>	pel:bookMark				
∑ Auto-Completion	<f11> ,</f11>	pel:auto-completion				
∑ Cut & Paste - Kill	<f11> -</f11>	pel:kill	Kill (cut) operations			
<u></u> Marking	<f11> .</f11>	pel:mark				
· ∑ Comments · ∑ Hide/Show	<f11> ;</f11>	pel:comment				
<u>∑ Cut &amp; Paste</u> - Copy	<f11> =</f11>	pel:copy	Copy operations			
	<f11> ?</f11>	pel:help				
	<f11> ? a</f11>	pel:apropos				
	<f11> ? d</f11>	pel:describe				
<u>∑ Help/Info</u>	<f11> ? e</f11>	pel:emacs				
	<f11> ? i</f11>	pel:info				

Content	<u>Operation</u>	<u>Keystroke</u>	Key Map	<u>Note</u>
Contact Position   Contact Pos		<f11> ? k</f11>	pel:keys	
Clark Annex	<u></u> File-mngt	<f11> B</f11>	pel:browse	Directory tree browsing (for now: it will evolve)
	∑ Cut & Paste - OS Clipboard	<f11> C</f11>	pel:clipboard	
Finance		<f11> D</f11>	pel:draw	
Section		<f11> D u</f11>	pel:plantuml	
Tags - Orios References	≫ Frames	<f11> F</f11>	pel:frame	
Figure   Section   Secti		<f11> S</f11>	pel:session	
Encentrol Turn - underlining		<f11> X</f11>	pel:xref	
Eachers			pel:underline	Underline text with specified character.
Emitters   411 b b   palsoniture			pel:abbrev	
Entitle			·	
Moderation			·	
Counting   Citi's			·	
Conta Margo		<f11> c</f11>		Counting text elements in current buffer
Contact Alexance   Cill   d   c   patrille   File & directory management			·	
Fine-mont   Sile A directory management   Sile A directory manag			·	
**Files ** Files ** F			·	File & directory management
File-monta	• <u>∑M Dired</u>			
File/Directory Variables				
Time	<u></u> File-mngt		·	
Timestring Text			·	
Techboard Macros   centeros   cf11> k   pelabmacros   pelamacros   pelamacros   pelamacros   pelamacros   cf11> k   pelamacros   pelamacros   cf11> k   pelamacros   cf11> k   pelamacros   pelamacros   cf11> k   pelamacros   pelamacros   cf11> k   pelamacros   pelamacros   pelamacros   cf11> k   pelamacros   pelama				
\$ \times \$\text{\$	∑ Inserting Text		·	
Total Pace of Semanton         fill > k 1         petermannes           Dissipar-Lines <f11> 1         petilineour           Coursor         <f11> n         petilineour           Scorting         <f11> n         petimouror         Multiple cursor editing.           Scorting         <f11> n         petimouror         Multiple cursor editing.           Scorting         <f11> c         petimouror         Multiple cursor editing.           Elegisters         <f11> c         petimouror         Correct edition.           Elegisters         <f11> c         petimouror         Correct fills.           Elegisters         <f11> c         petimouror         Petit fill and petimouror           Elimacional cursor         <f11> c         petitex -microscope           Elimacional cursor         <f11> c         petitex -microscope           Elimacional cursor         c         petitex -microsco</f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11>	∑ Keyboard Macros		·	Emacs keyboard macros, centimacro, emacros, elmacros.
∑ Display - Lines			·	
E Cursor			·	
∑ Sorting <f11> o         pebroder         Ordering/Sorting.           ∑ Registers         <f11> s         pebregister         February (f11) s         petregister           ∑ Search/Replace         <f11> s         pebregary (f11) s         f11&gt; s         pebregary (f11) s           ∑ Toxt Modes         <f11> t         pebregary (f11) s         petrograph           ∑ Toxt Modes         <f11> t         petrograph         Ferrograph           ∑ Toxt Modes         <f11> t         petrograph         Ferrograph           ∑ Flat Modes         <f11> t         petrograph         Ferrograph           ∑ Flat Modes         <f11> t         petrograph         Ferrograph           ∑ Flat Modes         <f11> t         petrograph         Ferrograph           ∑ Transpose         <f11> t         petrograph         Ferrograph           ∑ Windows         <f11> t         petrograph         Ferrograph           ∑ Windows         <f11> t         petrograph         Ferrograph           ∑ Windows         <f11> v         petrograph         Ferrograph           ∑ Windows         <f11> v         petrograph         Ferrograph           ∑ Sorolling         <f11> v         petrograph         Vanishow-size           &lt;</f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11>			·	
Executive <fil></fil>				,
Search/Replace			<u> </u>	Ordering/Sorting.
Search/Replace	<u>» Registers</u>			
Search/Replace   Cfl1> s w   pelsearch-word				
STAIM   Pelitext   Pelitext	∑ Search/Replace	-		
Search Modes			<u> </u>	
Salign	~~			
Settling/Justification   Cfil's t f   pelfill   Text fill			•	
Filling/Justification   cfl1> t j pel:justification   Text justification   Text justification	<u>» Align</u>			Tout fill
∑ Text Modes <f11> t m         peltext-modes           ∑ Transpose         <f11> t t         peltext-transpose           ∑ Whitespace             ∑ Undo/Redo/Repeat/Arg         <f11> v         peltvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         pelvos/dedicated         PEL also supports Git, a page dedicated for Git is not yet written           ∑ Windows         <f11> v         v         Pelvox Gut Als</f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11>	∑ Filling/Justification		·	
∑Transpose <fil> t t       peltext-transpose         ∑Whitespace       <fil> t w       peltext-whitespace         ∑Undo/Redo/Repeat/Arg       <fil> v       petundo         ∑VGS-Mercurial       <fil> v       pelvos       PEL also supports Git, a page dedicated for Git is not yet written         ∑Windows       <fil> v       pelvindow         ∑Windows       <fil> v d       pelwindow-dedicated         ∑Windows       <fil> v       pelwindow-dedicated         ∑Nelwis       <fil> v       pelwindow-size         ∑Inserting Text       <fil> v       y       pelyasnippet       Yasnippet text template insertion/expansion.         ∑Customize          pelyasnippet       Yasnippet text template insertion/expansion.         ∑Customize              ∑Customize              ∑Customize       <t< td=""><th>W Tout Modes</th><td></td><td>1</td><td>Text Justification</td></t<></fil></fil></fil></fil></fil></fil></fil></fil></fil></fil></fil></fil></fil></fil>	W Tout Modes		1	Text Justification
Whitespace   <f11> t w   pel:text-whitespace    </f11>			<u>'</u>	
∑ Undo/Redo/Repeat/Arg <fill> u       pelundo         ∑ VCS-Mercurial       <fill> v       pelvcs       PEL also supports Git, a page dedicated for Git is not yet written         ∑ Windows       <fill> w       pelwindow         ∑ Windows       <fill> w       pelwindow-dedicated         ∑ Windows       <fill> w       pelwindow-size         ∑ Shells       <fill> x       pelwanippet       Yasnippet text template insertion/expansion.         ∑ Scrolling       <fill>         pelsoroll         ∑ Scrolling       <fill> fill&gt; (f2&gt;       pelcifg         &lt; fill&gt; <fill> f2&gt;       pelcifg-pel-lang         &lt; fill&gt; <fill> f2&gt; E       pelcifg-pel-lang         &lt; fill&gt; <fill> f8&gt;       pelcifg-pel         ∑ Projectile          ∑ Menus       <fill> <fill> f8&gt;       pel:projectile         ∑ Speedbar       <fill> M-s       pel:projectile         X percolotic key maps       Pel:provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.       No e set has a key prefix that uses <fil> SPC followed by a key identifying the language.       The key prefix, simulating a local mode prefix.       The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.         XII def. AppleScript       &lt; fill&gt; 5PL&lt;</fil></fill></fill></fill></fill></fill></fill></fill></fill></fill></fill></fill></fill></fill></fill>				
∑ VCS-Mercurial <fill> v       pelvos       PEL also supports Git, a page dedicated for Git is not yet written         ∑ Windows       <fill> w       pelvwindow         ∑ Windows       <fill> w       pelvwindow-dedicated         ∑ Windows       <fill> w       pelvwindow-size         ∑ Shells       <fill> x       pelvasnippet       Yasnippet text template insertion/expansion.         ∑ Scrolling       <fill>         pelsoroll         ∑ Customize       <fill> fil&gt; cf2&gt; pelctfg       pelctfg         <fill> fil&gt; cf2&gt; pelctfg       cfil&gt; cf2&gt; pelctfg-pel-lang       cfil&gt; cf2&gt; pelctfg-pel         ∑ Projectile       cfil&gt; cf2&gt; p       pelctfg-pel         ∑ Menus       cfil&gt; cf10&gt; pelmenu       pelmenu         ∑ Speedbar       cfil&gt; ks       pelspeedbar         Major mode specific key maps       PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.       One set has a key prefix that uses x£11&gt; SPC followed by a key identifying the language.       The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.       The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.         ¾(- AppleScript       &lt; cf11&gt; SPC       pel:for-applescript   <th></th><td></td><td>· · · · · · · · · · · · · · · · · · ·</td><td></td></f12></fill></fill></fill></fill></fill></fill></fill></fill>			· · · · · · · · · · · · · · · · · · ·	
∑ Windows <f11> w       pel:window         ∑ Windows       <f11> w d       pel:window-dedicated         ∑ Windows       <f11> w s       pel:window-size         ∑ Shells       <f11> x       pel:eXecute         ∑ Inserting Text       <f11> y       pel:pasnippet       Yasnippet text template insertion/expansion.         ∑ Scrolling       <f11> f2&gt;       pel:croll          <f11> <f2>       pel:crg         <f11> <f2> pel:cfg          <f11> <f2> pel:cfg-pel-lang          <f11> <f2> pel:cfg-pel-lang          <f11> <f2> pel:cfg-pel          ∑ Menus       <f11> <f8>       pel:projectile         ∑ Menus       <f11> <f10>       pel:menu         ∑ Speedbar       <f11> M-s       pel:speedbar         Major mode specific key maps       PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.         * One set has a key prefix that uses <f11> SPC followed by a key identifying the language.       * The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.       * The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.         \$\frac{1}{2} \sqrt{1}</f12></f11></f11></f10></f11></f8></f11></f2></f11></f2></f11></f2></f11></f2></f11></f2></f11></f11></f11></f11></f11></f11></f11>			<u> </u>	PEL also supports Git a page dedicated for Git is not yet written
Windows   Cf11> w d   pel:window-dedicated	<u> </u>		<u> </u>	Supposed Sin, a page dealected for Git is not yet written
Swindows   <fill> w s   pel:window-size    </fill>			•	
Shells  Shell				
∑ Inserting Text				
∑ Scrolling <f11>         pel:scroll         ∑ Customize       <f11> <f2> SPC       pel:cfg-pel-lang          <f11> <f2> E       pel:cfg-pel-lang          <f11> <f2> P       pel:cfg-pel         ∑ Menus       <f11> <f8>       pel:projectile         ∑ Menus       <f11> <f10>       pel:menu         ∑ Speedbar       <f11> M-s       pel:speedbar         Major mode specific key maps       PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.         • One set has a key prefix that uses <f11> SPC followed by a key identifying the language.       • The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.         The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.         3)16-AppleScript       • <f11> SPC a       pel:for-applescript</f11></f12></f11></f11></f10></f11></f8></f11></f2></f11></f2></f11></f2></f11></f11>				Yasnippet text template insertion/expansion.
Customize   Cf11> <f2> SPC   pel:cfg    </f2>				
S Customize			·	
Section   Sect	<u> ▼ Customize</u>			
∑ Projectile <f11> <f8>       pel:projectile         ∑ Menus       <f11> <f10>       pel:menu         ∑ Speedbar       <f11> M−s       pel:speedbar         Major mode specific key maps       PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.       <ul> <li>One set has a key prefix that uses <f11> spc followed by a key identifying the language.</f11></li> <li>The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.</f12></li> <li>The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.</li> </ul>        ¾I € - AppleScript     <f11> spc a     pel:for-applescript</f11></f11></f10></f11></f8></f11>				
∑ Menus <f11> <f10>       pel:menu         ∑ Speedbar       <f11> M-s       pel:speedbar         Major mode specific key maps       PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.       One set has a key prefix that uses <f11> SPC followed by a key identifying the language.         The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.         The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.         Plt - AppleScript       <f11> SPC a       pel:for-applescript</f11></f12></f11></f11></f10></f11>	∑ Projectile			
∑ Speedbar <f11> M-s       pel:speedbar         Major mode specific key maps       PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.       <ul> <li>One set has a key prefix that uses <f11> spc followed by a key identifying the language.</f11></li> <li>The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.</f12></li> <li>The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.</li> </ul>            Plt-AppleScript</f11>			1 1	
Major mode specific key maps  PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.  One set has a key prefix that uses <f11> SPC followed by a key identifying the language.  The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.  The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.  PEL provides a set of global key-maps that are specific to major modes for markup and programming languages. The key maps have 2 set of bindings.  The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.  The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.</f12></f12></f11>		-		
bindings.  One set has a key prefix that uses <f11> spc followed by a key identifying the language.  The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.  The following list is ordered by programming languages names (sorting all Lisp under L) and then listing the markup languages after.  Ple-AppleScript  Pel:for-applescript</f12></f11>			· ·	for markup and programming languages. The key maps have 2 set of
PIE-AppleScript  • <f11> SPC a pel:for-applescript</f11>		<ul> <li>bindings.</li> <li>One set has a key prefix that uses <f11> SPC followed by a key identifying the language.</f11></li> <li>The other set is only available inside buffers that use the specific major mode and they all use the same <f12> key prefix, simulating a local mode prefix.</f12></li> </ul>		
7/4 / / / / / / / / / / / / / / / / / /	มีเ <b>ต้-</b> AppleScript	-		, , , , , , , , , , , , , , , , , , , ,
		• <f12></f12>		

### Compression	<u>Operation</u>	<u>Keystroke</u>	Key Map	<u>Note</u>
	<u> ФІ - С</u>		pel:for-c	
	<u>₩I - C</u> - C pre-processor		pel:for-c-propoc	
	<u>ℜɪ - C</u> - C tempo skeleton		pel:c-skel	Prefix for tempo skeletons for the C programming language.
-     -	<u> ФІ - С++</u>		pel:for-c++	
### Finance Lisp - compile   cf.112   spc. 1   peldor-ellutr   ### Finance Lisp - compile   cf.112   spc. 1   peldor-ellutr   #### Finance Lisp - compile   cf.112   spc. 1   peldor-ellutr   ##### Finance Lisp - compile   cf.112   spc. 1   peldor-ellutr   ####################################	<u>₩I - C++</u> - C pre-processor		pel:for-c++-preproc	
## Filing	<u> ФІ - D</u>		pel:for-d	
## Efriang - clause	<u>βί - Elixir</u>		pel:for-elixir	
* - (112	Bί - Erlang		pel:for-erlang	
	भ्रा - Erlang		pel:erlang-analysis	Planned
	भ्रा - Erlang - clause		pel:erlang-clause	
	भ्रा - Erlang - debug		pel:erlang-debug	
	भ्रा - Erlang - functions		pel:erlang-function	
	<u> ֆլ - Erlang</u> - tempo skeletons		pel:erlang-skel	Prefix for tempo skeletons for the Erlang programming language.
## - Common Lisp	Bι - Forth		pel:for-forth	
### - Emacs Lisp   - <f12>   - <f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12></f12>	Bt - Julia		pel:for-julia	
CFI2>   CFI2	भ्रा - Common Lisp		pel:for-lisp	
Spi - Emacs Lisp - compile   Spi   c	<u>‡</u> βι - Emacs Lisp		pel:for-elisp	
CF112   C   C   C   C   C   C   C   C   C	<u>⊈</u> Pι - Emacs Lisp - analyze		pel:elisp-analyze	
* * * * * * * * * * * * * * * * * * *	<u>‡βι - Emacs Lisp</u> - compile		pel:elisp-compile	
*   *   *   *   *   *   *   *   *   *	<u>‡βι - Emacs Lisp</u> - debug		pel:elisp-debug	
Comparison   Com	<u>‡βι - Emacs Lisp</u> - eval		pel:elisp-eval	
• <f12> 1</f12>	<u>‡βι - Emacs Lisp</u> - function		pel:elisp-function	
Skeletons   October   SPC p   Pel:for-python	ұ取 - Emacs Lisp - library		pel:elisp-lib	
• <f12>                                      </f12>			pel:elisp-skel	
• <f12>                                      </f12>	<b>β</b> ί - Python		pel:for-python	
• <f12>  M PlantUML  • <f11> SPC u • <f12>  M reStructuredText  • <f12>  M reStructuredText - adorn style • <f11> SPC r • <f12>  M reStructuredText - tempo  • <f11> SPC r • <f12>  Pel:for-reST   pel:for-rst-adorn  pel:for-rst-adorn  pel:for-rst-adorn</f12></f11></f12></f11></f12></f12></f11></f12>	Bι - REXX		pel:for-rexx	
• <f12> <u>M reStructuredText</u> • <f11> SPC r • <f12> <u>M reStructuredText</u> - adorn style • <f11> SPC r A • <f12> A  <u>M reStructuredText</u> - tempo • <f11> SPC r <f12> pel:for-rst-adorn  <u>M reStructuredText</u> - tempo • <f11> SPC r <f12> pel:for-rst-skel</f12></f11></f12></f11></f12></f11></f12></f11></f12>	M Graphviz Dot		pel:for-graphviz-dot	
• <f12>  M reStructuredText - adorn style • <f11> SPC r A • <f12> A  M reStructuredText - tempo • <f11> SPC r <f12> pel:for-rst-adorn  Planned</f12></f11></f12></f11></f12>	M PlantUML		pel:for-plantuml	
• <f12> A  M reStructuredText - tempo • <f11> SPC r <f12> pel:for-rst-skel  Planned</f12></f11></f12>	M reStructuredText		pel:for-reST	
<u>M reStructuredText</u> - tempo	M reStructuredText - adorn style		pel:for-rst-adorn	
- NILO NILO	M reStructuredText - tempo skeletons	• <f11> SPC r <f12> • <f12> <f12></f12></f12></f12></f11>	pel:for-rst-skel	Planned
Other Function Keys  PEL also uses the function keys for other purpose.  See the <u>Keys - Fn</u> table: it describes PEL's use of the functions keys with and without key modifiers.	Other Function Keys			with and without key modifiers.
Move point to next visible bookmark <f2>       (bm-next)       Not a prefix, a command: Move point to next visible bookmark.         Activated only when pel-use-bm is set to t. See ∑ Bookmarks.</f2>		<f2></f2>	(bm-next)	
Repeat last operation (repeat REPEAT-ARG)  Not a prefix, a command: Repeat most recently executed command.  See <u>\(\int\) Undo/Redo/Repeat/Arg</u>		<f5></f5>		, , ,
Text Insertion <f6> pel:f6</f6>	Text Insertion	<f6></f6>	pel:f6	
PEL Hydras  The head of all PEL Hydras. Activated on first use. The PEL Hydras are described in:  • 31.6- AppleScript • > Hide/Show • > Windows	PEL Hydras	<f7></f7>	PEL Hydras	The PEL Hydras are described in:  •   •   •   •   •   •   •   •   •   •
<u>▶ Projectile</u> <f8> projectile-command-map Activated by <f11> <f8> <f8> when pel-use-projectile is set to activate projectile.</f8></f8></f11></f8>	<u>» Projectile</u>	<f8></f8>	projectile-command-map	, , ,