Getting Help / Apropos / Descriptions / Info Manuals / Queries

		-	ptions / Info Manuals / Queries
Description Cotting Holp	Keystroke Emacs is a heavily document	Function Ited All of this documentation	Note is accessible from within Emacs: the manuals, the info page, the docstrings of functions
Getting Help Emacs Built-in Help describe, apropos info keys, Helpful, Log programming, extra, about man, woman Emacs bugs report PEL PDF Help With Emacs under SSH ►	and variables, the customize PEL supplements Emacs • See the <u>>Index</u> PDF v • These PDFs have a lar • Type the <f12> <f1: <f11="" by="" default="" github="" oper="" pdf="" pel="" raw="" the="" type="" •=""> ?</f1:></f12>	ation system. You can search for help with a large set of topic—with has links to all PEL PDFs. ge set of hyper links to each ot key sequence to open help for its the local PDF file using the local stead by changing the values <f2> to open their customization.</f2>	or manual, topic, command, function, variable, object names, values inside variables. specific PDF files such as this one (identified as **\sum_Help/Info*). her, to Emacs manual and external package home and description sites. or several major modes or <f11> ? p for help on a specific topic. acal PDF rendering application. You can force PEL to launch your default browser and open of of pel-flip-help-pdf-arg or and/or pel-open-pdf-method user-options.</f11>
Open this PDF file.	• <f11> ? <f1> • <f11> ? k <f1></f1></f11></f1></f11>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the <u>Nelp/Info</u> local PDF. If the prefix argument (like C-u or M) is used, then it opens the remote GitHub hosted raw PDF instead. If the pel-flip-help-pdf-arg useroption is set it's the other way around.
<u>∑ Customize</u> PEL Help Support	• <f11> ? <f2> • <f11> ? k <f2></f2></f11></f2></f11>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL help support and syntax tools. • If OTHER-WINDOW is non-nil (use C - u), display in other window.
<u>∑ Customize</u> Emacs Help Support	• <f11> ? <f3> • <f11> ? k <f3></f3></f11></f3></f11>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs grep support. Groups: apropos, command-log, debbugs, help, helpful, hydra, interaction-log, keycast, man, which-func, which-key.
Emacs Reference Cards		and to open the local copy of th	Access customization group with <f11> ? <f2></f2></f11> ese files if they are present. entify it in the pel-emacs-refcard-dirpath user option.
Open local copy of <u>Emacs</u> <u>PDF reference card</u>	<f11> ? e r</f11>	(pel-open-emacs-refcard)	Prompt for an Emacs REFCARD and open it. Supports tab completion. • Attempts to find the directory where the Emacs PDF reference card files are stored. Otherwise uses the directory identified by the pel-emacs-refcard-dirpath user option.
Emacs Help System	As described above, Emacs	provides help for almost every	thing. The list of commands to access this information is shown in the following rows.
Prefix Keys	Key sequences consist of ei	ther one keystroke like C-a or	M-b, or a key sequence that starts with a prefix, like C-x s, where C-x is the key prefix.
List all keys that belong to a prefix	<pre>• <prefix> C-h • <prefix> <f1></f1></prefix></prefix></pre>		Type C-h (or <f1></f1>) after the prefix keystroke to list all key bindings that belong to that prefix. For example to list all C-x r keys, type C-x r C-h
Describe Help			the command requests. The information is displayed in a read-only *Help* buffer.
Show all key commands for this buffer	• C-h b • <f1> b</f1>	(describe-bindings &optional PREFIX BUFFER)	Display a buffer showing a list of all defined keys, and their definitions. The keys are displayed in order of precedence.
Help on key binding	• C-h k <keys> • <f1> k <keys></keys></f1></keys>	(describe-key &optional KEY UNTRANSLATED UP- EVENT)	Display documentation of the function invoked by KEY. KEY can be any kind of a key sequence; it can include keyboard events, mouse events, and/or menu events. Get binding for the typed <keystroke> in the current context. Displays the name of the command function, it's description, it's bindings.</keystroke>
Open Info manual describing the command for the specific key	• C-h K <keys> • <f1> K <keys></keys></f1></keys>	(Info-goto-emacs-key- command-node KEY)	Open the info node in the Emacs manual which describes the command bound to KEY. • Interactively, if the binding is 'execute-extended-command', a command is read. • The command is found by looking up in Emacs manual's indices or in another manual found via COMMAND's 'info-file' property or the variable 'Info-file-list-for-emacs'
Print name of function invoked by key	• C-h c <keys> • <f1> c <keys></keys></f1></keys>	(describe-key-briefly &optional KEY INSERT UNTRANSLATED)	Print the name of the function KEY invokes. KEY is a string.
Describe active major/ minor(s) modes and the key bindings	• C-h m • <f1> m • <f11> ? k m</f11></f1>	(describe-mode &optional BUFFER)	Lists the active major mode, all active minor modes and the bound keystrokes.
Describe a package	• C-h P • <f1> P</f1>	(describe-package PACKAGE)	Display the full documentation of PACKAGE (a symbol). • Prompts for the package name, supports completion.
See also: <u>∑ Packages</u>		. 7.6.0.02	Shows whether it is installed or not, its version, the features it implements and some extra notes. Accesses the the elpa-compliant sites and downloads text file description.
Describe a function	• C-h f • <f1> f</f1>	(describe-function FUNCTION)	Display the full documentation of <u>FUNCTION</u> (a symbol). - For example: C-h f *-mode : Get a completion list of all emacs modes - The buffer shown contains link to the file where the function is implemented. Following the link will open the file in a buffer, even if the file is compressed.
Describe symbol	• C-h o • <f1> o</f1>	(describe-symbol SYMBOL &optional BUFFER FRAME)	Display the full documentation of SYMBOL. Will show the info of SYMBOL as a function, variable, and/or face.
Describe variable	• C-h v • <f1> v</f1>	(describe-variable VARIABLE &optional BUFFER FRAME)	- For example: C-h v load-path: shows the emacs lisp path Reference: https://www.gnu.org/software/emacs/manual/html_node/eintr/See-variable-current-value.html
Describe bindings for a command	• C-h w • <f1> w</f1>	(where-is DEFINITION & optional INSERT)	Print message listing key sequences that invoke the command DEFINITION. • Prompt for command name, supports completion. • If INSERT (the prefix arg) is non-nil, insert the message in the buffer
Help on Input Method	• C-h I • <f1> I</f1>	(describe-input-method INPUT-METHOD)	Provide information about the <u>input method</u> . Prompts for the name of an input method. See Input Method section for more info.
See also: <u>Number Input Method</u> Describe encoding system	• C-h C-\ • C-h C	(describe-coding-system	Display information about CODING-SYSTEM.
Describe buffers encoding ►	• <f1> C • <f1> C • <f11> ? d C</f11></f1></f1>	CODING-SYSTEM)	Prompts for coding system name. Supports completion. Type RET to describe current buffer encoding.
Describe language environment	• C-h L • <f1> L</f1>	(describe-language- environment LANGUAGE- NAME)	Describe how Emacs supports language environment LANGUAGE-NAME. • Prompts for language name, proposing the currently used language as the default. • Supports completion.
Describe <u>syntax-table</u> of current major mode	• C-h s • <f1> s</f1>	(describe-syntax &optional BUFFER)	Describe the syntax specifications in the syntax table of BUFFER. The descriptions are inserted in a help buffer, which is then displayed. BUFFER defaults to the current buffer. See also: Syntax Table @ Emacs Wiki
Show character syntax info and text properties	<f11> ? e .</f11>	(pel-syntax-at-point)	Display complete information for character at point. Opens a *Help* buffer to show extended character info and display text properties identified by the pel-syntax-text-properties user-option in the message area. Access with <f11>? <f2></f2></f11>
Emacs Apropos			ion in buffers using the info reader format. The info reader mode commands are shown an always get help on the current mode, that applies to the info reader mode as well.
Show information available about specified pattern	<f11> ? a a</f11>	(apropos PATTERN &optional DO-ALL)	Show all meaningful Lisp symbols whose names match PATTERN. Symbols are shown if they are defined as functions, variables, or faces, or if they have nonempty property lists. PATTERN can be a word, list of words (separated by spaces), or regexp (using some
			regexp special characters). For a word, search for matches for that word as a substring. For a list of words, search for matches for any two (or more) of those words.

<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>
Get a-propos info on command	• C-h a • <f1> a • <f11> ? a c</f11></f1>	(apropos-command PATTERN & optional DO-ALL VAR-PREDICATE)	Show commands (interactively callable functions) that match PATTERN. • With C-u prefix, or if 'apropos-do-all' is non-nil, also show non interactive functions. Old Emacs command name was: command-apropos.
	matches for that word as	a substring. If it is a list of wor	paces), or a regexp (using some regexp special characters). If it is a word, search for ds, search for matches for any two (or more) of those words. Emacs session, showing their key bindings and a quick description.
Look for topic in all info documents	<f11> ? i a</f11>	(info-apropos STRING)	Prompts for a string and looks up for that string in all the indices of all the Info documents installed in the system. Opens an Apropos index menu with the links to the found topics. Use this to <i>find the manual section(s) that describe a specific function or variable</i> .
Open the Info Reader on specific topic	• C-h i • <f1> i • <f11> ? i i • %-?</f11></f1>	(info &optional FILE-OR- NODE BUFFER)	Open the *info* buffer if already opened. If not, open the info reader for the top node. • A non-numeric prefix argument (C-u) directs this command to read a file name from the minibuffer. It is possible to open a compressed .info.gz file directly! Emacs will uncompress it and open it. • A numeric prefix argument of N selects an Info buffer named "*info* <n>".</n>
			y specify an Info node of the form "(FILENAME)NODENAME". owing actions available once emacs is in the Info Reader Mode.
Search for text in function and variables doc strings	• C-h d • <f1> d • <f11> ? a d</f11></f1>	(apropos-documentation PATTERN & optional DO- ALL)	Search for functions and variables whose documentation strings match the specified pattern and display the appropriate info pages.
List variables and functions defined in Emacs Lisp file.	<f11> ? a L</f11>	(apropos-library FILE)	List the variables and functions defined by library FILE. FILE should be one of the libraries currently loaded and should thus be found in 'load-history'.
Show buffer-local variables	<f11> ? a l</f11>	(apropos-local-variable PATTERN &optional BUFFER)	Show buffer-local variables that match PATTERN. Optional arg BUFFER (default: current buffer) is the buffer to check.
Show user option	<f11> ? a o</f11>	(apropos-user-option PATTERN &optional DO- ALL)	Show user options that match PATTERN. PATTERN can be a word, a list of words (separated by spaces), or a regexp (using some regexp special characters). If it is a word, search for matches for that word as a substring. If it is a list of words, search for matches for any two (or more) of those words. • With C-u prefix, also show variables, not just user options.
Show all symbols that have a specific value	<f11> ? a u</f11>	(apropos-value PATTERN & optional DO-ALL)	Show all symbols whose value's printed representation matches PATTERN. PATTERN can be a word, a list of words (separated by spaces), or a regexp (using some regexp special characters). If it is a word, search for matches for that word as a substring. If it is a list of words, search for matches for any two (or more) of those words. With C-u prefix, or if 'apropos-do-all' is non-nil, also looks at function definitions (arguments, documentation and body) and at the names and values of properties.
Show variables that match a specific name pattern	<f11> ? a v</f11>	(apropos-variable PATTERN &optional DO- NOT-ALL)	Show variables that match PATTERN. • With the optional argument DO-NOT-ALL non-nil (or when called interactively with the prefix C-u), show user options only, i.e. behave like 'apropos-user-option'.
Emacs Info		Ī	
Open specified info manual	<f11> ? i m</f11>	(info-display-manual MANUAL)	Prompt for a specific Info manual to open in a buffer. • Example: "eintr" := Introduction to Emacs Lisp.
Open Emacs Manual describing a specified command function	• C-h F • <f1> F</f1>	(Info-goto-emacs- command-node COMMAND)	Go to the Info node in the Emacs manual for command COMMAND. • The command is found by looking up in Emacs manual's indices or in another manual found via COMMAND's 'info-file' property or the variable 'Info-file-list-for-emacs'. COMMAND must be a symbol or string.
Find specified function function or variable in info	• C-h S • <f1> F</f1>	(info-lookup-symbol SYMBOL &optional MODE)	Display the definition of SYMBOL, as found in the relevant info manual. • When this command is called interactively, it reads SYMBOL from the minibuffer. In the minibuffer, use M-n to yank the default argument value into the minibuffer so you can edit it. The default symbol is the one found at point. • With prefix arg MODE a query for the symbol help mode is offered.
Info reader mode keys	The keys that can be typed in the "Info" buffers and their meanings include the following: ? Get Info help ? Page down into the node text, move to following text/node if already at end ? Page Down ? Page up into the node text, move to following text/node if already at end ? Page Up into the node text, move to previous text/node if already at top ? Page up into the node text, move to previous text/node if already at top ? Page up into the node text, move to previous text/node if already at top ? Page up into the node text, move to other node) t. Move to the top of the Info document . Next node in the current level . Move to the top of the Info document . Next node in the current level . With ace-link external package to a clivated when the pel-use-ace-link: highlight each target with a target key Previous node (any level) . Previous Node (any level) . Next Node (any level) . Next Node (any level) . Info History, wish instory forward . Menu - enter nodes sub-menu (at cursor position) . Menu - enter nodes sub-menu (at cursor position) . Menu - enter nodes sub-menu (at cursor position) . Type a number between 1 to 9 to select the corresponding menu entry. 1 := first Menu - enter nodes sub-menu (at cursor position) . Type a number between 1 to 9 to select the corresponding menu entry Menu/Cross-Reference - Move cursor to nodes inext sub-menu/cross-reference link . Menu/Cross-Reference - Move cursor to nodes inext sub-menu/cross-reference link . Menu/Cross-Reference - Move cursor to nodes inext sub-menu/cross-reference link . Menu/Cross-Reference - Move cursor to nodes inext sub-menu/cross-reference link . Menu/Cross-Reference - Move cursor to nodes inext sub-menu/cross-reference link . Menu/Cross-Reference - Move cursor to nodes inext sub-menu/cross-reference link . Search Info - search inext search and <ret> . To repeat search serve in the fine for a string.</ret>		

<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>
Key Sequence help			clearly show. Key strokes are extended in various ways and key prefixes is one of them. g the key sequences, list the remaining available bindings, and list recent history of the
List command history See also: Undo/Redo/Repeat/Arg	<f11> ? d H</f11>	(list-command-history)	List history of commands that used the minibuffer. • Show list of commands in the *Command History* buffer as a list of Emacs Lisp forms.
Toggle which-key mode	<f11> ? k K</f11>	(which-key-mode &optional ARG)	Toggle which-key-mode. When which-key mode is enabled, and you type a prefix key, all keys bound following this prefix are shown in the mini buffer (if you wait long enough to let them display). Requires the which-key package. PEL activates it when pel-use-which-key is t.
Show state of PEL numlock	<f11> ? k #</f11>	(pel-show-mac-numlock)	Display state of 'pel-mac-keypad-numlocked' used to control the numeric keypad.
Show state of key-chord mode. See: <u>∑ Key-Chords</u>	• <f11> <f5> k ? • <f11> ? k M-K</f11></f5></f11>	(pel-key-chord-describe)	Show state of key-chord-mode. When key-chord mode is on, list key chord bindings in a help buffer.
Show top level bindings in the map of the current major mode	<f11> ? k k</f11>	(which-key-show-major- mode)	Show top-level bindings in the map of the current major mode. This function will also detect evil bindings made using 'evil-define-key' in this map. These bindings will depend on the current evil state. Requires the which-key package. PEL activates it when pel-use-which-key is t.
Toggle keycast mode on/off	<f11> ? k c</f11>	(keycast-mode &optional ARG)	Show current command and its key binding in the mode line. Use it to create a screen cast to show how to use Emacs. This requires the keycast external package PEL makes keycast available when the pel-use-keycast user option is set to t.
Show personal key bindings	<f11> ? k b</f11>	(describe-personal- keybindings)	Display all the personal keybindings defined by 'bind-key'.
Display free keys	<f11> ? k f</f11>	(free-keys &optional PREFIX	Display free keys in current buffer.
	You can change the prefix	sequence by hitting 'p' in the	A free key is a key without associated key-binding as determined by 'key-binding'. quence are considered, possibly together with modifier keys from 'free-keys-modifiers'. *Free keys* buffer. Prefix is supplied in format recognized by 'kbd', for example "C-x". nis when the pel-use-free-keys user option is t.
Display last few typed characters	• C-h 1 • <f1> 1 • <f11> ? k 1</f11></f1>	(view-lossage)	Display last few input keystrokes and the commands run. To record all your input, use 'open-dribble-file'.
Record ALL typed characters to a file	M-x open-dribble- file	(open-dribble-file FILE)	Start writing all keyboard characters to a dribble file called FILE. • If FILE is nil, close any open dribble file. • The file will be closed when Emacs exits. • Be aware that this records all characters you type! Don't type passwords at that time!
Redo/edit last complex command executed	• C-x Esc Esc • C-x M-Esc	(repeat-complex-command ARG)	Edit and re-evaluate last complex command, or ARGth from last. • A complex command is one which used the minibuffer. The command is placed in the
See also: <u>Vundo/Redo/</u> Repeat/Arg	• C-x M-:		 minibuffer as a Lisp form for editing. The result is executed, repeating the command as changed. If the command has been changed or is not the most recent previous command it is added to the front of the command history. You can use the minibuffer history commands M-n and M-p to get different commands to edit and resubmit.
Helpful - extended help for Emacs with more contextual information	The helpful external package provides the same help information provided by Emacs with more contextual information and extra links. This requires the helpful external package PEL installs and activates it when the pel-use-helpful user-option is set. These commands provide a lot more information than standard Emacs help. Use then to debug, trace, look at references, etc		
Help for function/macro/ special form	<f1> <f2> a</f2></f1>	(helpful-callable SYMBOL)	Show help for function, macro or special form named SYMBOL.
Help for command	<f1> <f2> c</f2></f1>	(helpful-command SYMBOL)	Show help for interactive function named SYMBOL.
Help for function	<f1> <f2> f</f2></f1>	(helpful-function SYMBOL)	Show help for function named SYMBOL.
Help for key	<f1> <f2> k</f2></f1>	(helpful-key KEY- SEQUENCE)	Show help for interactive command bound to KEY-SEQUENCE.
Help for macro	<f1> <f2> m</f2></f1>	(helpful-macro SYMBOL)	Show help for macro named SYMBOL.
Help for symbol	<f1> <f2> o</f2></f1>	(helpful-symbol SYMBOL)	Show help for SYMBOL, a variable, function or macro.
Help for variable	<f1> <f2> v</f2></f1>	(helpful-variable SYMBOL)	Show help for variable named SYMBOL.
Help for symbol at point	<f1> <f2> .</f2></f1>	(helpful-at-point)	Show help for the symbol at point.
Log keys & commands	PEL provides access to two different packages you can use to show the commands and their key bindings as you type them • These can be used to show what you type during a presentation to other users, or for documentation purpose. The following 2 external packages are supported: The command-log-mode external package. PEL activates it when the pel-use-command-log-mode user option is turned on (set to t). PEL activates it when the pel-use-interaction-log-mode user option is turned on (set to t).		
Command Log Mode	The information is similar to • See the <u>∑ Windows</u> table	what is available with view-loss for commands that can be use	ows the log of all key sequence and mouse events and the executed command name. sage, but in a nicely formatted way, much easier to use. ed to toggle the dedicated state of the window allowing you to move the window.
	 This requires the command-log-mode.el file from the command-log-mode external package. •		
Toggle command logging for current buffer	<f11> ? k c c</f11>	(command-log-mode &optional ARG)	Toggle command logging: command-log-mode in the current buffer. • The command-log lighter is shown on the mode line while the minor mode is active.
Toggle command logging for all buffers	<f11> ? k c C</f11>	(global-command-log- mode &optional ARG)	Toggle command logging globally: for all buffers. The command-log lighter is shown on the mode line while the minor mode is active.
Open Command Log buffer	<f11> ? k c o</f11>	(clm/open-command-log- buffer &optional ARG)	Opens (and creates, if non-existant) a buffer used for logging keyboard commands. • With any prefix argument, the existing command log buffer is cleared.
Close Command Log buffer	<f11> ? k c .</f11>	(clm/close-command-log- buffer)	Close the command log window. Logging continues while the window is closed.
Toggle log of all commands	<f11> ? k c /</f11>	(clm/toggle-log-all)	Toggle the logging of all commands: activate/de-activate common command filtering. • command-log-mode either logs all commands or filter some often used ones like the cursor and character movements. The default setting is controlled by the clm/log-all. • The list of non-logged commands is controlled by clm/non-logged-commands.
			9

Description	<u>Keystroke</u>	Function	Note
Interaction Log Mode	The interaction-log externa		mand-log-mode shown above, but more powerful. It shows the key bindings, the Emacs
		nserted text and other informate a separate Emacs frame allow	ion in different colours. ing you to continue showing information even after using C-x 1 to maximize the current
	window.	·	rits author: Torstein Krause Johansen.
	The interaction-log extension		activates it when the pel-use-interaction-log-mode user option is turned on (set to t).
Start/stop interaction log	<f11> ? k i i</f11>	(interaction-log-mode	Global minor mode logging keys, commands, file loads and messages.
mode		&optional ARG)	 Logged information goes to the *Emacs Log* buffer. On first invocation the buffer is created but not shown. Select it or use the command pel-interaction-log-buffer to show it.
Show interaction log buffer	<f11> ? k i b</f11>	(pel-interaction-log-buffer)	Show interaction log buffer.
Display interaction log in a separate frame.	<f11> ? k i f</f11>	(ilog-show-in-new-frame)	Display log in a pop up frame. Customize 'ilog-new-frame-parameters' to specify parameters of the newly created frame.
Toggle display of buffer names in the interaction log	<f11> ? k i n</f11>	(ilog-toggle-display-buffer- names)	Toggle display of buffers in log buffer for each key event. This command must be issued inside the interactive log buffer only.
Toggle interaction log view	<f11> ? k i v</f11>	(ilog-toggle-view)	Toggle between different view states: showing only messages, only commands, only file loads, and everything. • This command must be issued inside the interactive log buffer only.
Programming Help	PEL provides key bindings f	or the following commands that	tt are useful when editing source code files.
Show what completion mode is currently used.	<f11> M-c ?</f11>	(pel-show-active- completion-mode)	Display the completion mode currently used.
Toggle which-function-mode to display name of current	• <f11> ? f • <f11> M-d f</f11></f11>	(which-function-mode &optional ARG)	Toggle mode line display of current function (Which Function mode). • With a prefix argument ARG, enable Which Function mode if ARG is positive, and
function at point	The publish former	le ie e glebel min 1 11"	disable it otherwise. en enabled, the current function name is continuously displayed in the mode line.
See also: • <u>> Menus</u> • <u>> Mode Line</u>	Detection of functions a	nd variables depend on the ime	en enabled, the current function hame is continuously displayed in the mode line. enu functionality. If you modify the content of a buffer, you need to force a menu rescan to -rescan, bound to <f11> <f10> r.</f10></f11>
	ldentify the major modes		in the which-function-mode user-option with M-x customize-option which-
	function-mode. • With PEL you can use:		
	,		ation group. Note that it will provide access to the customization group even when the
		ch-function-mode RET to a	access the user-option directly.
Show syntax of char at point	<f11> ? d s</f11>	(pel-show-char-syntax)	Display a message showing the character syntax of character at point.
Extra Descriptions	PEL implements a set of ext	ra commands and bindings to	built-in Emacs commands to display other the following extra information.
Show symbols of currently active major mode	<f11> ? ?</f11>	(pel-show-major-mode)	Display the symbol of the currently active major mode.
Show which search tool is currently used	<f1> ? s</f1>	(pel-show-active-search-tool)	Display the currently used search tool.
Show available colours	<f11> ? d c</f11>	(list-colors-display &optional LIST BUFFER- NAME CALLBACK	Display names of defined colors, and show what they look like.
Show encoding of file visited in current buffer • See also: The Help/Info	<f11> ? d e</f11>	(pel-show-buffer-file- encoding)	Show coding system of file in current buffer. • Open a *Help* buffer and show the value of the buffer-file-coding-system variable.
List all available faces	<f11> ? d F</f11>	(list-faces-display &optional REGEXP)	List all faces, using the same sample text in each.
Show buffer and file name	<f11> ? d f</f11>	(pel-show-window- filename-or-buffer-name)	Show the (full path) name of the file or buffer of current window.
Show information about an input method	<f11> ? d i</f11>	(list-input-methods)	Display information about all input methods.
Display content of kill ring	<f11> ? d k</f11>	(pel-show-kill-ring)	Display content of 'kill-ring' in *Help* buffer.
Print current buffer line # (and narrowed line #)	<f11> ? d 1</f11>	(what-line)	Print the current buffer line number and narrowed line number of point.
Query info about point Show information about	• C-x = • <f11> ? d p</f11>	(what-cursor-position &optional DETAIL)	Displays information about character at point in the echo area: position, character, encoding.
current character.	''	·	how the complete information of character at point with all properties, face, encoding, etc.
	• Type: C-u C-x =		With PEL, you can also type: C C-x =
Show window dimension	<f11> ? d w</f11>	(pel-show-window-sizes)	Show the height & width of the current window.
Display ASCII table	<f11> ? A</f11>	(ascii-table)	Show an interactive ASCII table in the other (next) window. Requires the <u>ascii-table</u> package
See also: <u>Number Input Method</u>			PEL activates this when the pel-use-ascii-table user option is t .
About Emacs	Information about Emacs, it	s environment and configuratio	n is available through a set of commands listed below
Display Emacs version	<f11> ? e v</f11>	(emacs-version)	Display Emacs version
Display Emacs uptime	<f11> ? e u</f11>	(emacs-uptime &optional FORMAT)	Display a string giving the uptime of this instance of Emacs in the echo area.
Open local copy of Emacs PDF reference card	<f11> ? e r</f11>	(pel-open-emacs-refcard)	Prompt for an Emacs REFCARD and open it. Supports tab completion
PDF reference card	1	tory where the Emacs PDF refe	erence card files are stored. Failing to detect them, dit uses the directory identified by the
Show number of available and key bound commands	<f11> ? e c</f11>	(pel-emacs-command- stats)	Display number of available commands and the number of those that have key bindings in the echo area, and the number of bindings in the global map.
Show loaded files & features	<f11> ? e 1</f11>	(pel-emacs-load-stats &optional WITH_DETAILS)	Display the number of loaded files and the number of features currently loaded. • With C-u prefix print features in a buffer. With C-u C-u , also print load information,
Display Memory Usage	<f11> ? e m</f11>	(pel-emacs-mem-stats)	with symbols displayed as clickable buttons that open a help buffer describing it. Display Emacs memory statistics inside an *emacs-mem-stats* buffer.
Check/display list of shadowed Emacs Lisp files	<f11> ? e s</f11>	(list-load-path-shadows &optional STRINGP)	Display a list of Emacs Lisp files that shadow other files • Shows any shadows in a '*Shadows*' buffer
Print imenu controlling variables	<f11> ? e i</f11>	(pel-imenu-print-vars)	Print the value of the imenu variables used to control the imenu functionality for the current buffer. Symbols are clickable buttons to help on the symbol.
See also: <u>∑ Menus</u>			 Print this information in a *imenu-dbg* buffer. Use to investigate the imenu support for a major mode.

Description	<u>Keystroke</u>	Function	<u>Note</u>	
Print value of outline controlling variables See also: <u>Noutline</u>	<f11> ? e o</f11>	(pel-outline-print-vars)	Print the current buffer specific values of outline controlling variables. Use this to learn possible how to control the outline minor mode.	
See Emacs executable path	<f11> ? e x</f11>	(pel-emacs-executable)	Display Emacs executable path in echo area.	
Display load-path	<f11> ? e p</f11>	(pel-emacs-load-path & optional N)	Show the current load-path inside a new *load-path* buffer. Open the buffer in the current window or the one identified by N, with the display-line-number-mode on. The buffer is NOT committed to a file.	
	Window selection: If N If N If N If N If N	is not specified, nil or 1: open to is negative, create a new winds is 0: copen buffer in other in [2,8] range, open buffer in with a := 'up 4 := 'left 5 := 'current 6 : 2 := 'down	ow and open buffer inside it. er window indow identified by the direction corresponding to the cursor in a numeric keypad: = 'right	
Display Emacs initialization	• <f11> ? e t</f11>	is 9 or larger: search in window (pel-show-init-time)	Display benchmark startup time.	
time with benchmark information if available		itialization and duration tree in a side the echo area. -init library to measure time of ge, select benchmark-init and ille and place the following lines rk Measurement	2 buffers if the benchmark-init library is installed and loaded in the init.el file. It also display the various loaded modules. I install it. s as close as possible to the top of the file:	
	;; CAUTION: Modi: (require 'benchm' (expand'''-/.emi (add-hook 'after-	d-file-name acs.d/elpa/benchmark-init- -init-hook 'benchmark-init	rsion is available20150905.938/benchmark-init"))	
List processes	• Update the path in this co	(list-processes &optional	Display a list of all processes that are Emacs sub-processes.	
See also: <u>∑ Shells</u>	• <f11> ? e C-p • <f11> z ?</f11></f11>	QUERY-ONLY BUFFER)	With non-nil optional argument, only processes with the query-on-exit flag set are listed. Any process listed as exited or signalled is actually eliminated after the listing is made.	
ESUP - Emacs Start Up Profiler	<f11> ? e P</f11>	(esup &optional INIT-FILE &rest ARGS)	Profile the startup time of Emacs in the background. If INIT-FILE is non-nil, profile that instead of USER-INIT-FILE. ARGS is a list of extra command line arguments to pass to Emacs.	
	Requires the esup extern	nal package. 🛂 PEL activates	it when the $\textbf{pel-use-esup}$ customization variable is set to $\textbf{t}.$	
	1		orts Emacs running in graphics mode. 2) esup steps into 'require' and 'load' forms at the tatements. This limits its usefulness when conditional loading is located in the init.el file and	
	when the use-package mac	ros are used. Both of these tec	chniques are used by PEL to reduce init time.	
Emacs + PEL specifics	Ü	ovide more information about E		
Show PEL user option and package info See also: Customize	<f11> ? e ?</f11>	(pel-package-info &optional FULL-REPORT)	Display the following information in the echo area: The number of PEL user-options, and the number of them that are active. The number of Elpa packages active: the count of the ones directly installed because of active PEL user-options and the count of them installed as dependencies of the first group. The number of Emacs Lisp files stored in the ~/.emacs.d/utils (or equivalent directory) as a result of PEL user options.	
Display name of	• <f11> ? e <f2></f2></f11>	(pel-setup-info-dual-	 With optional argument, generates a full report with much more details in a *pel-user-options* report buffer. Any key prefix works. M <f11>? e ? for example.</f11> Display current PEL customization setup. 	
customization file. Show whether PEL dual independent customization is used or not. See also: Customize	• <f11> <f2> ?</f2></f11>	environment)	 Check two independent customization files for terminal/tty and graphics mode are requested and if so check if they are setup properly. Report an error and list problems if there are any, otherwise display the current setup. After executing that command you will have to edit your init.el file and set the pel-use-graphic-specific-custom-file-p symbol to t. 	
Display current Emacs Startup configuration setup See also: Fast Startup	• <f11> ? e M-S • <f11> M-S ?</f11></f11>	(pel-setup-info)	Display current state of PEL setup: whether Emacs startup is used in normal or in fast startup operation mode.	
Using Man inside Emacs See also: • %1 - Erlang • © Customize	Emacs provide 2 main commands to display man pages inside buffers. Both of these are much more powerful than the usual man reader available on the shell allowing navigation across man pages & opening hyperlinks. The man command uses the system man utility, while woman is a complete implementation which has some formatting limitations compared to man but it's very useful in systems where man is not available. The man command will find pages that the system's man can find. This can be extended or modified by setting the MANPATH environment variable. Inside Emacs you can also customize the Emacs Man-switches user option to provide extra configuration including a different MANPATH by using the -M switch. For an example see how to add Erlang man pages in the \$\mathbf{N}\cappa \in \text{Erlang table}.			
Open a man page inside an Emacs buffer	• <f11> ? m • <m-f8> • %-M</m-f8></f11>	(man MAN-ARGS)	Open a Man page inside an Emacs window.	
On Unix/Linux, use it to display help about C/C++ functions, types.	Using man pages inside emacs is even better than using it from the shell because: • The links are active and can be followed. When the man page describes a directory or file, emacs will open the file or the directory (in direct mode) when pressing <ret> over the link. • You can navigate easily between sections (n/p will move to the next/previous section). You can use any of the searches. • You can use any of the options to the man command at the prompt, like the -a option to access all man pages of the same name. Then use M-n and M-p to move from one to the other page, inside the same buffer. • See all keys available in mode, with <f1> m or <f11>? k m. • The man command prompts, using the word at point as the default. • PEL key sequence to customize man: <f11> <f2> E m • The man command provides completion at prompt. However, if you set up a MANPATH to isolate on directory to get only the list of commands in a specified set of man pages (eg. for Erlang commands only), the completion will only work if the man directory contains a whatsis database file. See my description on how to create whatis file for local man directory.</f2></f11></f11></f1></ret>			
Open man page for item at point	<m-s-f8></m-s-f8>	(pel-man-at-point)	Open a man page for the topic at point if any, otherwise prompts for topic. • Man page section controlled by user option named pel-%s-man-section, where '%s' is replaced by the major mode. Useful for modes like Tcl where section name differs.	
Open a man page without external man process: woman	• <f11> ? w • <c-f8></c-f8></f11>	(woman &optional TOPIC RE-CACHE)	Open a man page file in Emacs using the woman mode, completely implemented in Emacs Lisp (and therefore without using the external 'man' process).	
	That can be very useful under environments where man is not available (such as basic Microsoft Windows ®). PEL key sequence to customize woman: <f11> <f2> E w With ace-link external package activated when the pel-use-ace-link user option is set to t., the following key is activated:</f2></f11>			
		on: highlight each target with a		

<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>
Emacs Bug Reports See also: EmacsBugTracker@	The <u>GNU Bug Tracker</u> isMore info is available in the	used as a bug tracker for seve e GNU Bug Tracker Docume	
Emacs Wiki Emacs Bug triaging article	PEL activates it whe	n the pel-use-debbugs user o	es by using the debbugs external package. Applion is turned on (set to t). PEL also binds the debbugs commands to the following keys. Per via the <f11>? <f3> key sequence.</f3></f11>
List all outstanding Emacs bugs	<f11> ? b a</f11>	(debbugs-gnu SEVERITIES &optional PACKAGES ARCHIVEDP SUPPRESS TAGS)	List all outstanding bugs.
Search for Emacs bugs	<f11> ? b s</f11>	(debbugs-gnu-search PHRASE &optional QUERY SEVERITIES PACKAGES ARCHIVEDP)	Search for Emacs bugs interactively. Search arguments are requested interactively. The "search phrase" is used for full text search in the bugs database. Further key-value pairs are requested until an empty key is returned. If a key cannot be queried by a SOAP request, it is marked as "client-side filter". When using interactively, use C-x M-: after this command for reusing the argument list. Be careful in editing the arguments, because the allowed attributes for QUERY depend on PHRASE being a string, or nil. See Info node '(debbugs-ug) Searching Bugs'.
List all users tags	<f11> ? b u</f11>	(debbugs-gnu-usertags &rest USERS)	List all user tags for USERS, which is ("emacs") by default.
List bug reports that contain a patch	<f11> ? b p</f11>	(debbugs-gnu-patches)	List the bug reports that have been marked as containing a patch.
List all bugs or specified bugs	<f11> ? b b</f11>	(debbugs-gnu-bugs &rest BUGS)	List all BUGS, a list of bug numbers. In interactive calls, prompt for a comma separated list of bugs or bug ranges, with default to 'debbugs-gnu-default-bug-number-list'. This accepts a single bug number, a comma separated list of bug numbers as well as dash separated range of bug numbers.
List bugs tags locally	<f11> ? b t</f11>	(debbugs-gnu-tagged)	List the bug reports that have been tagged locally.
List all outstanding Emacs bugs in Org-mode format	<f11> ? b A</f11>	(debbugs-org)	List all outstanding bugs using an Org-mode format.
Search for Emacs bugs, list bugs in Org-mode format	<f11> ? b S</f11>	(debbugs-org-search)	Search for bugs interactively. List bugs in Org-mode format. Search arguments are requested interactively. The "search phrase" is used for full text search in the bugs database. Further key-value pairs are requested until an empty key is returned. If a key cannot be queried by a SOAP request, it is marked as "client-side filter".
List bug reports that contain a patch, list bugs in Org- mode format	<f11> ? b P</f11>	(debbugs-org-patches)	List the bug reports that have been marked as containing a patch. List bugs in Org-mode format.
List all bugs or specified bugs in Org-mode format	<f11> ? b B</f11>	(debbugs-org-bugs)	List all bugs, a list of bug numbers. List bugs in Org-mode format. • In interactive calls, prompt for a comma separated list of bugs or bug ranges, with default to 'debbugs-gnu-default-bug-number-list'.
List bugs tags locally in Org- mode format	<f11> ? b T</f11>	(debbugs-org-tagged)	List the bug reports that have been tagged locally. List bugs in Org-mode format.
More Help			
Open Emacs Tutorial	• C-h t • <f1> t</f1>	(help-with-tutorial &optional ARG DONT-ASK- FOR-REVERT)	Open an Emacs Tutorial. Restore location if used before (after prompt).
Find Elisp Package See also: <u>Nackages</u>	• C-h p • <f1> p</f1>	(finder-by-keyword)	Find packages matching a given keyword. Useful to search for packages supporting a specific concept.
Open Emacs FAQ	• C-h C-f • <f1> C-f</f1>	(view-emacs-FAQ)	Display the Emacs Frequently Asked Questions (FAQ) file.
Emacs news	• C-h n • <f1> n</f1>	(view-emacs-news &optional VERSION)	Display info on recent changes to Emacs. With argument, display info only for the selected version. Includes code modifications of each version of Emacs.
Display local help in echo area	<fl>. C-h . C-c ! H</fl>	(display-local-help &optional ARG)	Display local help in the echo area. This displays a short help message, namely the string produced by the 'kbd-help' property at point. If 'kbd-help' does not produce a string, but the 'help-echo' property does, then that string is printed instead. A numeric argument ARG prevents display of a message in case there is no help. While ARG can be used interactively, it is mainly meant for use from Lisp.
Open PEL PDF Help File	· ·	DF files such as this one for seitHub hosted raw copy of these	veral topics. e help PDF using the PEL help commands bound to the <f1> key available inside several</f1>
See also: <u>➤Legend</u>	PEL key prefixes. • For example several major modes bind the <f12><f1> key sequence to PEL help. Other such as help for abbreviations is <f11> a <f1> • The complete list of these key bindings is listed below. • Unfortunately not all Help PDF files have key sequences for them. However, you can: • Open any PDF file with the pel-help-pdf-select command: it prompts for a topic with tab completion support: use <f11> ? p • Open a dired buffer on the local directory where all PDF files are stored with <f11> ? P . Select the file(s) and type z to open the selected file(s). ■ When opening a GitHub hosted raw PDF you will want to use a browser that is capable of rendering the PDF directly instead of downloading the file to your file-system. Firefox does this by default. You may need to install a plugin for other browsers. ■ These PEL functions select the local PDF or the remote GitHub raw PDF via a combination of factors: ■ By default it opens the local PDF file unless a prefix argument is specified by typing C-u or M before the command key sequence. ■ If the pel-flip-help-pdf-arg user-option is set to t, it's the other way around: it opens the remote GitHub raw PDF by default. ■ By default, when opening a remote GitHub raw PDF it uses the default browse method selected by the browse-url-browser-function user-option. ■ The default value of the browse-url-browser-function user-option is to use the system browser. ■ You can force another one by setting PEL's pel-browser-used user-option to 'firefox or 'chrome. ■ When you do that the new selected browser is also used for the local file. If you want to use this browser for remote files only and keep Emacs' ability to use a PDF viewer for local files, then set the pel-open-pdf-method user-option to 'pdf-viewer.</f11></f11></f1></f11></f1></f12>		
Open this PDF file.	<f11> ? <f1></f1></f11>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the <u>Nelp/Info</u> local PDF. If the prefix argument (like C-u or M) is used, then it opens the remote GitHub hosted raw PDF instead. If the pel-flip-help-pdf-arg useroption is set it's the other way around.
Select and Open a PEL PDF file	• <f11> ? p • <f11> p</f11></f11>	(pel-help-pdf-select &optional OPEN-WEB- PAGE)	Prompt for a PEL PDF and open it. By default it opens the local PDF file, but if the OPEN-WEB-PAGE argument is non-nil it opens the web-based PDF copy hosted on Github. Supports tab completion.
Open a Dired Buffer for PEL PDF files.	<f11> ? P</f11>	(pel-help-pdfs-dir)	Open a Dired buffer on the PEL PDF directory. Inside Dired you can open a PDF file by typing 'z' over the file name. You can also select several and type 'z' to open them all.

D	Manager I.a.	Fig. 15:0
<u>Description</u>	<u>Keystroke</u>	Function Note
<u>≻Index</u>	<f11> <f1></f1></f11>	Open <u>➤ Index</u> PDF file, a quick index with links to all other PEL PDF files.
∑ Abbreviations	<f11> a <f1></f1></f11>	Open <u>Nabbreviations</u> PDF file.
<u>∑ Align</u>	<f11> t a <f1></f1></f11>	Open <u>S Align</u> PDF file.
<u>Natio-Completion</u>	<f11> , <f1></f1></f11>	Open <u>National Auto-Completion</u> PDF file.
<u>∑ Bookmarks</u>	<f11> ' <f1></f1></f11>	Open <u>Sookmarks</u> PDF file.
<u>∑ Buffers</u>	<f11> b <f1></f1></f11>	Open <u>S Buffers</u> PDF file.
∑ Case Conversions	<f11> t <f1> 1</f1></f11>	Open <u>Case Conversions</u> PDF file.
<u>∑ Comments</u>	<f11> ; <f1></f1></f11>	Open <u>S Comments</u> PDF file.
<u> ∑ Cut & Paste</u>	• <f11> = <f1> • <f11> - <f1></f1></f11></f1></f11>	Open <u>Nature</u> Cut & Paste PDF file.
∑ Counting	<f11> c <f1></f1></f11>	Open <u>S</u> Counting PDF file.
<u> ∑ Cursor</u>	<f11> m <f1></f1></f11>	Open <u>S Cursor</u> PDF file.
∑ Customize	<f11> <f2> <f1></f1></f2></f11>	Open <u>S Customize</u> PDF file.
➤ Diff & Merge	<f11> d <f1></f1></f11>	Open Diff & Merge PDF file.
∑ Dired	<f11> f <f1> 2</f1></f11>	Open Dired PDF file.
∑ Drawing	<f11> D <f1></f1></f11>	Open > Drawing PDF file.
<u>▼ Enriched Text</u>	<f11> t e <f1></f1></f11>	Open <u>Senriched Text</u> PDF file.
<u>∑ Fast Startup</u>	<f11> <f2> \$ <f1></f1></f2></f11>	Open the <u>Fast Startup</u> PDF file.
<u>∑ File-mngt</u>	<f11> f <f1> 1</f1></f11>	Open <u>S File-mngt</u> PDF file.
∑ File/Directory Variables	<f11> f v <f1></f1></f11>	Open <u>File/Directory Variables</u> PDF file.
∑ Filling/Justification	<pre>• <f11> t f <f1> • <f11> t j <f1></f1></f11></f1></f11></pre>	Open <u>Filling/Justification</u> PDF file.
<u>∑ Frames</u>	<f11> F <f1></f1></f11>	Open <u>Frames</u> PDF file.
<u></u> ∑ Grep	<f11> g <f1></f1></f11>	Open <u>S Grep</u> PDF file.
<u>∑ Help/Info</u>	<f11> ? <f1></f1></f11>	Open <u>Nelp/Info</u> PDF file.
<u></u> Hide/Show	<f11> M-/ <f1></f1></f11>	Open <u>N Hide/Show</u> PDF file.
∑ Highlight	<f11> h <f1></f1></f11>	Open <u>Neighlight</u> PDF file.
∑ Indentation	<f11> TAB <f1></f1></f11>	Open <u>Nation</u> PDF file.
∑ Input Method	<f11> t <f1> 2</f1></f11>	Open <u>∑ Input Method</u> PDF file.
∑ Inserting Text	• <f11> i <f1></f1></f11>	Open <u>∑ Inserting Text</u> PDF file.
<u> </u>	• <f11> y <f1> • <f11> y <f1> • <f11> _ <f1></f1></f11></f1></f11></f1></f11>	Open <u>// Inserting Text</u> PDF line.
∑ Keyboard Macros	<f11> k <f1></f1></f11>	Open <u>Neyboard Macros</u> PDF file.
∑ Key-Chords	<f11> <f5> k <f1></f1></f5></f11>	Open the <u>Ney-Chords</u> PDF file.
Line management. <u>∑ Display - Lines</u>	<f11> 1 <f1></f1></f11>	Open <u>National Display - Lines PDF file.</u>
<u></u> Marking	<f11> . <f1></f1></f11>	Open <u>Narking</u> PDF file.
<u></u> Mode Line	<f11> M-d <f1></f1></f11>	Open <u>Mode Line</u> PDF file.
<u></u> Menus	<f11> <f10> <f1></f1></f10></f11>	Open <u>Menus</u> PDF file.
∑ Outline	<f11> M-1 <f1></f1></f11>	Open <u>S Outline</u> PDF file.
∑ Projectile	• <f11> <f8> <f1> • <f8> <f1></f1></f8></f1></f8></f11>	Open <u>▼ Projectile</u> PDF file. • The key sequence <f8> <f1> is available when the projectile mode is activated.</f1></f8>
	<f11> r <f1></f1></f11>	Open Name Registers PDF file.
∑ Scrolling	<f11> <f1></f1></f11>	Open S Scrolling PDF file.
∑ Search/Replace	<f11> s <f1></f1></f11>	Open S Search/Replace PDF file.
∑ Sessions	<f11> S <f1></f1></f11>	Open S Sessions PDF file.
∑ Shells	<f11> 3 <11></f11>	Open S Shells PDF file.
	<f11> 2 <f1> <f1> o <f1></f1></f1></f1></f11>	· -
Sorting Speedbar		Open Sorting PDF file (o for ordering).
Speedbar Speedbar	<f11> M-s <f1></f1></f11>	Open Speedbar PDF file.
∑ Spell Checking	<f11> \$ <f1></f1></f11>	Open Sepell Checking PDF file.
Text Modes	• <f11> t <f1> 3 • <f11> t m <f1></f1></f11></f1></f11>	Open <u>Nature</u> Text Modes PDF file.
∑ Time Tracking	<f11> T <f1></f1></f11>	Open <u>∑ Time Tracking</u> PDF file.
<u>Nanspose</u>	<f11> t t <f1></f1></f11>	Open <u>National Transpose</u> PDF file.
<u></u> Whitespace	<f11> t w <f1></f1></f11>	Open <u>Whitespace</u> PDF file.
∑ Undo/Redo/Repeat/Arg	<f11> u <f1></f1></f11>	Open <u>Nundo/Redo/Repeat/Arg</u> PDF file.
∑ VCS-Mercurial	<f11> v <f1></f1></f11>	Open <u>VCS-Mercurial</u> PDF file.
<u></u> Web	<f11> f <f1> 3</f1></f11>	Open <u>N Web</u> PDF file.
<u></u> <u>Windows</u>	<f11> w <f1></f1></f11>	Open <u>Windows</u> PDF file.
∑ Xref	<f11> x <f1></f1></f11>	Open <u>∑ Xref</u> PDF file.
Specialized Minor		or specific programming languages
Modes		
βί- Lispy	This is available for the Li	lobal key binding for Lispy. sp family languages as well as Julia and Python.

<u>Description</u>	<u>Keystroke</u>	Function Note	
Mode Specific PDF	PEL PDF files for specific m	ajor modes can be opened using the <f12> <f1> key from a buffer in that mode. Inside another mode the longer key</f1></f12>	
Help: • Programming Languages	sequence that starts with <	F11> SPC is available.	
ழு≰- AppleScript	<f11> SPC a <f1></f1></f11>	Open <u>\$1.6- AppleScript</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
	<f12> <f1></f1></f12>		
<u> рі - С</u>	<f11> SPC c <f1> <f12> <f1></f1></f12></f1></f11>	Open <u>B1 - C</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
β ῖ - C++	<f11> SPC C <f1></f1></f11>	Open % I - C++ PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
β Ι - Clojure	<f11> SPC C-j <f1> <f12> <f1></f1></f12></f1></f11>	Open <u>\$\mathbf{Y}_1\$ - Clojure</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
mr D	<f11> SPC D <f1></f1></f11>	Open βι - D PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command	
<u> </u>	<f12> <f1></f1></f12>	prefix (like C-u) was used.	
βί - Erlang	<f11> SPC e <f1></f1></f11>	Open <u>$\mathfrak{P}\mathfrak{l}$ - Erlang</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
<u>βι - Elixir</u>	<f11> SPC x <f1><f12> <f1></f1></f12></f1></f11>	Open <u>\text{91} - Elixir</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
Pĭ - Forth	<f11> SPC f <f1></f1></f11>	Open 31 - Forth PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
<u> </u> βί - Go	<f11> SPC g <f1></f1></f11>	Open <u>Ní - Go</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command	
	<f12> <f1></f1></f12>	prefix (like C-u) was used.	
<u> 1</u> βί - Hy	<f11> SPC C-h <f1></f1></f11>	Open <u>\$1 - Hy</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command	
	<f12> <f1></f1></f12>	prefix (like C-u) was used.	
भ्रा - Gleam	<f11> SPC M-G <f1><f12> <f1></f1></f12></f1></f11>	Open the <u>\$\Pi\left(- Gleam\)</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
取ι - Javascript	<f11> SPC i <f1></f1></f11>	Open 31 - Hy PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command	
	<f12> <f1></f1></f12>	prefix (like C-u) was used.	
भृर - Julia	<f11> SPC j <f1><f12> <f1></f1></f12></f1></f11>	Open <u>\mathbb{B}\tau\ - Julia</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
ฺฆั - Janet	<f11> SPC T <f1></f1></f11>	Open the %I - Janet local PDF. If the prefix argument (like C-u or M) is used, then it opens the remote GitHub	
pr - variet	<f12> <f1></f1></f12>	hosted raw PDF instead. If the pel-flip-help-pdf-arg user-option is set it's the other way around.	
<u></u> ⊈₩ι - Emacs Lisp	<f11> SPC 1 <f1></f1></f11>	Open **\frac{\psi \cdot \text{Emacs Lisp}}{2} PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C - u) was used.	
भ्रा - Common Lisp	<f11> SPC L <f1><f12> <f1></f1></f12></f1></f11>	Open <u>\mathfrak{Pl} - Common Lisp</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like $\mathbf{C} - \mathbf{u}$) was used.	
Bι - LFE	<f11> SPC C-1 <f1></f1></f11>	Open %1 - LFE PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
野ι - NetRexx	<f11> SPC N <f1></f1></f11>	Open 31 - NetRexx PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
野ι - Python	<f11> SPC p <f1></f1></f11>	Open <u>NI - Python</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
Bι - REXX	<f11> SPC R <f1><f12> <f1></f1></f12></f1></f11>	Open <u>\$1 - REXX</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
ழ் - Rust	<f11> SPC r <f1></f1></f11>	Open BI - Rust PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
βί - Scheme	<f11> SPC C-s <f1></f1></f11>	Open <u>\$1 - Scheme</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
2 2	<f12> <f1></f1></f12>		
क्रा - ∨ ***	<f11> SPC v <f1> <f12> <f1></f1></f12></f1></f11>	Open $\mathfrak{P}\mathfrak{l}$ - V PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like $C-u$) was used.	
Build Tools			
भ्रा - Make	<f11> SPC M <f1></f1></f11>	Open % I - Make PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a	
	<f12> <f1></f1></f12>	command prefix (like C-u) was used.	
Markup languages			
M Graphviz Dot	<f11> SPC M-g <f1></f1></f11>	Open M Graphviz Dot PDF using method specified by the pel-open-pdf-method user-option or the alternate one if	
	<f12> <f1></f1></f12>	a command prefix (like C-u) was used.	
M Outline/Org-Mode	<f11> SPC M-o <f1></f1></f11>		
	<f12> <f1></f1></f12>	if a command prefix (like C-u) was used.	
M PlantUML	• <f11> D u <f1> • <f11> SPC M-u</f11></f1></f11>	Open M PlantUML PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
	<f1></f1>		
	<f12> <f1></f1></f12>		

<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>
M Markdown	<f11> SPC M-m <f1></f1></f11>	Open <u>M Markdown</u> PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used.	
	<f12> <f1></f1></f12>		
<u>M</u> reStructuredText	<f11> SPC M-r <f1></f1></f11>	Open M reStructuredText PDF using method specified by the pel-open-pdf-method user-option or the alternate a command prefix (like C-u) was used.	
	<f12> <f1></f1></f12>		

Help - References

Topic & Link	Description
Emacs Help	
GNU Emacs Manuals Online	The page with the list of all available online GNU Emacs manuals.
GNU Emacs Manual - Help	Emacs manual - Help chapter
Gnu Emacs Manual - Help Mode	Describes the command and key bindings that can be used in the Help-mode buffer window, which shows the help information.
Emacs Manuals	Note that all Emacs manuals are available inside of Emacs. It's better to test, investigate code, etc
GNU Emacs Manuals Online	Lists all GNU Emacs manuals, reference cards, etc
GNU Emacs Manual	Points to different formats of the manual. The format where all is inside one HTML file is useful to search. There's also the PDF formats.
GNU Reference Cards	This is accessible via the first link.
Emacs Papers	
EMACS: The Extensible, Customizable Display Editor	This paper was written by Richard Stallman in 1981 and delivered in the ACM Conference on Text Processing.
Emacs Tutorials	
A Guided Tour of Emacs	The official Emacs Tutorial. Part of Emacs. Best used <i>inside</i> Emacs. A good starting point. Use the others to get different point of views.
Absolute Beginner's Guide to Emacs	
A Tutorial Introduction to GNU Emacs	
Practical Emacs Tutorial @ ErgoEmacs	
Emacs Cheat Sheet / Keystroke Lists	Note, however, that Emacs itself and PEL provides similar information.
Emacs Videos	
Emacs Rocks - home	A collection of Youtube homed videos about various Emacs features. Well documented with keystrokes showing on the screen cast. Worth watching slowly to catch what is being done.
Emacs and Man files	
How to create a local whatis file	Show how to create a missing whatis file for a set of man pages and the philosophy behind apropos, whatis and makewhatis.