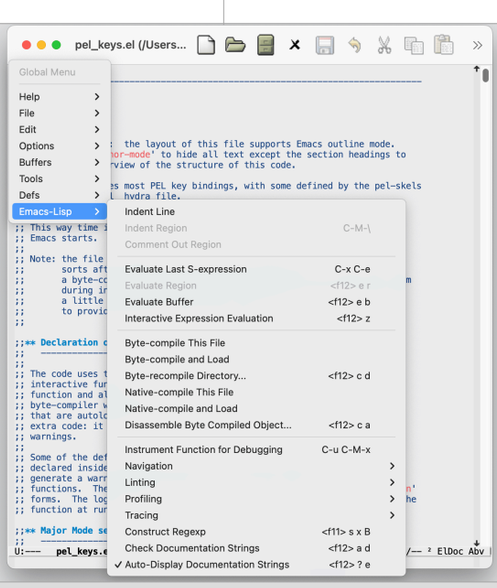
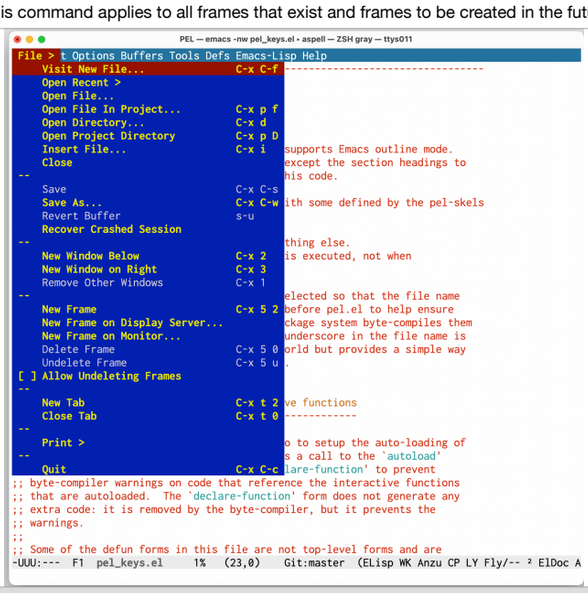
↖ Topic Index

Menus and iMenu

Description	Keystroke	Function	Notes / Example description

Emacs Menus
○ Help & Customization
• **Using Emacs Menubar**
  • **Emacs Buffer Menu**
• **Emacs iMenu**
  ○ find definitions using imenu
  • Control imenu behaviour
  • Show imenu behaviour. variables
  ○ Navigate to definitions via iMenu
 	
See also:
• Σ **Buffers**
• Σ **Completion/Input**
• Σ **Navigation**
• Σ **Speedbar**
	Emacs has several **top-level general purpose menus**:
	1. **Emacs MenuBar** is accessible when Emacs is running in graphics mode as well as when it is running in text terminal mode.
	• In graphics mode, Emacs MenuBar (the global menu) in the location controlled by the Operating System. For example, on macOS, the graphical mode Emacs Menu bar shows in the macOS menu bar. You can also open a local view of the MenuBar with **<f10>** the key; it shows overplayed on top of the current Emacs frame. You can also use the mouse.
	• In text terminal mode, Emacs MenuBar is displayed at the top of the Emacs frame when the **menu-bar-mode** is active (the default).
	  • Navigate the menu by typing the **<f10>** key. You can also use the mouse if it was enabled (via **<f11><f12>** in PEL). See Σ **Mouse** for information about using the mouse.
	  • If the **menu-bar-mode** is not active you can navigate the MenuBar in the ini buffer by typing the **<f10>** key.
	2. **Emacs Buffer Menu**, which list all buffers (see Σ **Buffers**). This opens locally, popping-up over the current frame when using the **C-<f10>** key.
	3. **Emacs iMenu** used to find major definitions in a file by their names. It shows information about the current buffer such as function names, etc...
	• The major mode of the current buffer must support imenu. See iMenu/Speedbar support and Speedbar/iMenu Mode Compatibility.
	• The iMenu system comes with a simple user interface providing tab-completion list.
	• The iMenu items can also be displayed in the Index section of the MenuBar and the BufferMenu.
	• The iMenu behaviour and user interface can be modified and extended by several external packages.
	PEL provides the following customization control which provides access to some of these packages:
	• **pel-imenu-follows-order-p** user-option controls whether entries are sorted or follows the order of declaration in the file.
	• **flimenu** external package activated by **pel-use-flimenu** user-option, controls whether iMenu lists are flatten or hierarchical.
	• **imenu+** external library activated by **pel-use-imenu+** user-option, activates and extends the basic iMenu MenuBar with a DEFS index entry that supports sorting options, ability to index commented definitions, bookmarks, etc.... ⚠ PEL uses **my imenu+ fork** which simplifies code by dropping support for Emacs < 25, and fixes all warnings and error similar to the one fixed in *patch for imenu+.el version 1145* .
	• **imenu-anywhere** external package activated by **pel-use-imenu-anywhere** user-option is used by **pel-goto-symbol-any-buffer** to jump to symbol definition of any buffer using one of the following input completion method. The user-option must be set to one of the following values:
	  • Use emacs-default: basic Emacs completion. Use tab to see possible matches.
	  • Use **Ido**. **pel-use-ido** must be turned on.
	  • Use **Ivy**. Requires **Ivy mode** **pel-use-ivy** must be on.
	  • Use **helm**. Requires **Helm mode** **pel-use-helm** must be turned on.
	• **imenu-extra** external package activated by **pel-use-imenu-extra** user-option, adds menu entries extracted from the major mode.
	• **popup-imenu** external package activated by **pel-use-popup-imenu** user-option, provides one pop-up menu for the iMenu content.
	• **popup-switcher** external package activated by **pel-use-popup-switcher** user-option, provides the same as popup-imenu and more.
	
	Last updated on: 2025-12-02		

Open this PDF file. See also: Σ **Help/Info**	**<f11> <f10> <f1>**	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the Σ **Menus** 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** user-option is set it's the other way around.
Σ **Customize** PEL imenu support	**<f11> <f10> <f2>**	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize PEL imenu support. Provides access to: • **pel-imenu-follows-order-p** **pel-use-flimenu** • **pel-use-imenu+** **pel-use-imenu-anywhere** • **pel-use-imenu-extra** **pel-use-popup-imenu** • **pel-use-popup-switcher** • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Σ **Customize** Emacs menu mechanism	**<f11> <f10> <f3>**	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs packages related to menu and imenu. Provides access to the customization of: • **menu** **imenu** **Imenu-Plus** **flimenu** • **popup-imenu** **popup-switcher** • When a prefix argument (like **C-u**) opens the buffer inside another window.
Using Emacs MenuBar	The MenuBar is shown when the menu-bar-mode is active. It is active by default. • In graphics mode, the MenuBar does not take any extra space, as opposed to the extra row at the top of the frame in text terminal mode. • If it is not active, hitting **<f10>** will open the menu inside the minibuffer are (as **<f11> <f10> t** does). • To activate or de-activate the menu-bar-mode, use the **<f11> C-<f10>** key stroke.		
Open main menu	**<f10>**	(**menu-bar-open** &optional FRAME)	Start key navigation of the menu bar in current frame. If **pel-use-imenu+** user-option is on, the index is always active under the DEFS menu entry. It also includes ability to toggle sorting, indexing commented definitions, and adds several categories in Emacs Lisp code.
Open main menu in Minibuffer	**<f11> <f10> t**	(**tmm-menubar** &optional X-POSITION)	Opens Emacs menu in the minibuffer instead of the graphical or text pop-up menu. Useful from the keyboard. ☛ ✂ The standard binding for this command is **M-`**. But PEL re-binds **M-`** to something else. ☛ Once the Index has been added to the menu (see below), you can use the **<f11><f10> i** keys to list the items in the minibuffer. The list provided is not shown in order though. The MenuBar Index section is listed in order.
Toggle MenuBar Mode	**<f11> <f10> B**	(**menu-bar-mode** &optional ARG)	Toggle display of a menu bar on each frame (Menu Bar mode). • With a prefix argument ARG, enable Menu Bar mode if ARG is positive, and disable it otherwise. • This command applies to all frames that exist and frames to be created in the future.
Main Menu in Emacs in Emacs running in Graphics mode and in terminal mode The main menu bar is shown in the two screen captures here: the graphics mode is shown to the left. This is the standard Emacs running in graphics mode under macOS. Emacs is heavily customizable and you could have an Emacs system that looks very different. However the menu is likely to be shown in the same way. Emacs running in terminal is shown to the right, with the main menu activated. Navigating the main menu is easier in graphics mode as sub-menus are displayed as expected. In terminal mode only one menu is shown at a time. Use the cursor keys to navigate in the text menu.			
Emacs Buffer Menu	The list of buffers is available via the Buffer popup-menu. It's also available via the buffer commands (see the Buffers table).		
Open buffer menu See also: Σ **Buffers**	• **C-<f10>** • **C-<down-mouse-1>**	(**buffer-menu-open**)	Start key navigation of the buffer menu: List buffers in a drop-down menu. • Lists buffers by major-mode. In graphics mode this can also be invoked using the **C-<down-mouse-1>**

Description	Keystroke	Function	Notes / Example description
Emacs IMenu, Index of items in buffer. See also: **iMenu/Speedbar support** which describes what is needed by a major mode to support iMenu.	Emacs IMenu (index menu) facility provides a menu of the file's items: • Programming language function definitions, type definitions, variables, etc… • Markup file elements, for example the list of document sections for a **reStructuredText** or **Markdown** text file. The imenu items can be displayed in several ways, with 3 of them controlled directly by the imenu library: • Under the Emacs MenuBar, as a drop-down menu under Index. • This must first be activated. To activate it hit **`<f11> <f10> I`**. Then hit **`<f10>`** to open the MenuBar and navigate to the Index entry. • Or one of the two representation, when the **imenu** command executes: • As a *completion buffer* (the default). Type the symbol you search and use tab completion. • Type **`tab tab`** to see the complete list. Hit **`return`** to select and move the point to the location of that item. • As a *pop-up menu*. With PEL. you can dynamically change the following imenu properties: • toggle between showing the imenu entries in a hierarchical fashion or a flat list (if the number of items is smaller than imenu max size) • toggle between entries listed alphabetically or in order of appearance in the file with **pel-imenu-toggle-follows-order** • toggle between using a completion buffer and a pop-up menu by executing the **pel-imenu-toggle-popup** The following commands allow you to use the iMenu mechanisms. • There's also commands you can use to investigate iMenu support while developing support for a major-mode.		
Add Buffer's IMenu to menu bar	**`<f11> <f10> I`**	(**imenu-add-menubar-index**)	Activates the Index entry in the Menu bar for the current buffer. The MenuBar Index lists functions, variables, types, etc.. • Once created, the index is available on the MenuBar. It is also becomes available via the following keystrokes: **`<f11><f10> i`** • It is, however, always available via the **`M-g i`** and **`M-g M-i`** keystrokes, therefore this command is not very useful.
Find definitions using IMenu See also: • Σ **Completion/Input** • Σ **Navigation**	• **`<f11> <f10> i`** • **`M-g i`** • **`M-g M-i`**	(**imenu** INDEX-ITEM)	Lists imenu-detected items from the current buffer (according to its major mode). • For example, in a elisp file, the entry points are the function definitions and may include the variables and other items depending what function does the parsing (it can be semantic which provides more information). Provides one of the following interfaces to let user select entry to jump to: • The default: input completion, using the minibuffer window and tab completion. • a pop-up window : available in Graphics mode selected by mouse or in both graphics and terminal (TTY) modes when the **imenu-use-popup-menu** user-option is turned on. • with PEL you can use **pel-imenu-toggle-popup** (bound to **`M-g <f4> p`**) to toggle the user interface used by **imenu**.
	☝ PEL provides the **pel-goto-symbol** bound to **`M-g h`**, to move point to the location of a symbol listed by i**menu** but also other user interfaces such as Ido, Ivy, Helm and some other popup menu mechanisms. See Σ **Completion/Input** and Σ **Navigation**.		
Toggle imenu between a hierarchical and a flat list.	• **`<f11> <f10> f`** • **`M-g <f4> f`**	(**pel-imenu-toggle-flatten**)	Toggles between imenu using a hierarchical menu (the default) and a flat menu. • Note that when the number of items to display exceeds the maximum length of the imenu, there imenu will be split anyway in multiple sections and will end up being "hierarchical" but instead of being split by type of content, it will be split on type and by alphabetical names. • 🎛 The maximum number of entries in a imenu list is controlled by 2 imenu user-options: • imenu-max-items: size limit of a pop-up imenu. • imenu-max-item-length: size limit of a drop down imenu 📦 Requires **flimenu** external package ☑ activated by **pel-use-flimenu** user-option.
Toggle order of appliance in the imenu	• **`<f11> <f10> o`** • **`M-g <f4> o`**	(**pel-imenu-toggle-follows-order**)	Changes the order of entries in the imenu between the default and the order of appearance of the symbols in the buffer. 🎛 Set the default with the **pel-imenu-index-follows-order-p** user-option.
Toggle imenu I/F between completion buffer and pop-up menu	• **`<f11> <f10> p`** • **`M-g <f4> p`**	(**pel-imenu-toggle-popup** &optional IN-CURRENT-BUFFER)	Toggle the use of pop-up menu versus completion buffer for imenu. • By default this applies to imenu issued in all buffers, but with the IN-CURRENT-BUFFER argument set the change applies only to the current buffer.
Toggle automatic imenu rescan	• **`<f11> <f10> R`** • **`M-g <f4> R`**	(**pel-imenu-toggle-auto-rescan**)	Toggle imenu automatic rescan • Default is set by **imenu-auto-rescan** user-option.
Force immediate imenu rescan	**`<f11> <f10> r`**	(**pel-imenu-rescan**)	Force imenu to immediately rescan the current buffer to find definitions.
Print imenu controlling variables See also: Σ **Help/Info**	**`<f11> ? e i`**	(**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. • Print this information in a *imenu-dbg* buffer. • Use this when investigating the imenu support for a major mode: use as a (currently primitive) Emacs development tool.

Display current setting of commands:
• **pel-goto-symbol**
• **pel-goto-symbol-any-buffer**
	`M-g ?`	(**pel-show-goto-symbol-settings**)	Display current settings used by the goto symbol commands in the echo area.
			Something like this:
			`goto-symbol UI is: popup-switcher`
			`goto-any-buffer UI is: Ido`
			`- iMenu lists are not flatten.`
			`- Ido uses:`
			`- Ido prompt geometry: grid mode, starts collapsed: expand with tab`
			`- Ido Ubiquitous mode: off`
			`- flx-ido mode: off`

Select Input Completion used by pel-goto-symbol	**`M-g <f4> h`**	(**pel-select-goto-symbol-UI**)	Select the input completion method used by the **pel-goto-symbol** command for the duration of the current editing session. • When Emacs starts the method used is determined by the value of the **pel-initial-goto-symbol-UI** user-option. You can use this command to change what is used in the current editing session without affecting the customized default. • See also the commands to control input completion (see Σ **Completion/Input**) • **pel-select-ido-geometry**: **`M-g <f4> M-g`** • **pel-ido-ubiquitous** : **`M-g <f4> M-u`** • **pel-flx-ido** : **`M-g <f4> M-f`**
Select Input Completion Method used by pel-imenu-anywhere	**`M-g <f4> y`**	(**pel-select-goto-symbol-any-buffer-UI**)	Select the input completion method used by the **pel-imenu-anywhere** command for the duration of the current editing session and used by the **pel-goto-symbol-any-buffer** command. • When Emacs starts the method used is determined by the value of the PEL **pel-use-imenu-anywhere** user-option. You can use this command to change what is used in the current editing session without affecting the customized default.

Description	Keystroke	Function	Notes / Example description

• **List and navigate to symbol definition**
  • **in *current* buffer**
  • **In all opened buffers**
 	
See also:
• Σ **Completion/Input**
• Σ **Navigation**
• Σ **Speedbar**
			The following command can be used to move point to any quickly selected a symbol definition, in any major mode supported by **Emacs imenu**.
			• Most major modes for programming and markup languages support imenu. PEL adds extra support for some modes.
			• PEL provides 2 commands:
			  • **pel-goto-symbol** lists target symbols in the current buffer, allowing you to select one and jump to it.
			  • **pel-goto-symbol-any-buffer** does the same but for all buffers currently opened.
			• For each of these commands PEL provides a selectable user interface. The user interface used for each command when Emacs starts is selected by a customization user-option variable. During an editing session PEL provides a UI selection command. In both cases the available user interfaces depend on what you activate.
			  • Customize **pel-goto-symbol** user interface with **`M-g <f4> <f2>`** to access the customization buffer:
			  • the **pel-initial-goto-symbol-UI** user option. Select one of:
			  • 0 = Use Emacs default: imenu
			  • 1 = Use Ido. Requires **idomenu** **pel-use-ido** and **pel-use-idomenu** must both be turned on.
			  • 2 = Use Ivy. Requires **Ivy mode** and **Ivy mode completion with Counsel mode** **pel-use-ivy** and **pel-use-counsel** must both be on.
			  • 3 = Use helm. Requires **Helm mode** **pel-use-helm** must be turned on.
			  • 4 = Use popup-imenu. Requires **popup-imenu** **pel-use-popup-imenu** to be turned on (in **pel-pkg-for-imenu** group).
			  • 5 = Use popup-switcher. Requires **popup-switcher** **pel-use-popup-switcher** to be turned on (in **pel-pkg-for-imenu** group).
			  • Modify the **pel-goto-symbol** UI for the current editing session with the **pel-select-goto-symbol-UI** command, bound to **`M-g <f4> h`**.
			
			  • Customize **pel-goto-symbol-any-buffer** user interface with with **`M-g <f4> <f2>`** to access the customization buffer:
			  • Requires **imenu-anywhere** **pel-use-imenu-anywhere** user option must be set to one of the following values:
			  • Use emacs-default: basic Emacs completion. Use tab to see possible matches.
			  • Use Ido. **pel-use-ido** must be turned on.
			  • Use Ivy. Requires **Ivy mode** **pel-use-ivy** must be on.
			  • Use helm. Requires **Helm mode** **pel-use-helm** must be turned on.
			  • Modify **pel-goto-symbol-any-buffer** UI for the current editing session with the **pel-select-goto-symbol-any-buffer-UI** command, bound to **`M-g <f4> y`**.
			• Use **pel-show-goto-symbol-settings ,** bound to **`M-g ?`** to show the current settings for both commands.
			When using Ido, for you have more options: you can select a different Ido prompt geometry and whether it uses 'flx' fuzzy matching.
			• Ido prompt geometries:
			  • The Emacs default: Ido linear selection,
			  • Grid initially collapsed or expanded. Requires **ido-grid-mode** Activate it with **pel-use-ido-grid-mode** user-option turned on.
			  • Vertical list. Requires **ido-vertical-mode** Activate it with **pel-use-ido-vertical-mode** user-option turned on.
			  • Select the initial geometry with the **pel-initial-ido-geometry**. Change it in the editing session with **pel-select-ido-geometry** (**`M-g <f4> M-g`**).
			• Ido 'flx' fuzzy matching requires **flx-ido**. Activate it with **pel-use-flx** user-option turned on.
			• Also use **`<f11> <f10> <f2>`** to customize the PEL iMenu user-options which have an impact on the way the iMenu entries are displayed.
			☝ Note that it is also possible to use the **Speedbar** (which also uses the symbols detected by **imenu**). See Σ **Speedbar** .

Find definitions using IMenu See also: • Σ **Completion/Input** • Σ **Navigation**	• **`<f11> <f10> i`** • **`M-g i`** • **`M-g M-i`**	(**imenu** INDEX-ITEM)	Lists imenu-detected items from the current buffer (according to its major mode). • For example, in a elisp file, the entry points are the function definitions and may include the variables and other items depending what function does the parsing (it can be semantic which provides more information). Provides one of the following interfaces to let user select entry to jump to: • The default: input completion, using the minibuffer window and tab completion. • a pop-up window : available in Graphics mode selected by mouse or in both graphics and terminal (TTY) modes when the **imenu-use-popup-menu** user-option is turned on. • with PEL you can use **pel-imenu-toggle-popup** (bound to **`M-g <f4> p`**) to toggle the user interface used by **imenu**.
Move point to imenu detected symbol definition in: current buffer ★★	• **`M-g h`** • **`M-g M-h`**	(**pel-goto-symbol**)	Prompt using for imenu symbol of the current buffer and move point to it. • Refresh imenu and jump to a place in the buffer using the completion method selected. • Modify user interface currently used with **`M-g <f4> h`**. • The command sets a ref-marker before moving. Return to previous location by typing **`M-,`**
Move point to imenu detected symbol definition of: all opened buffers ★★	• **`M-g y`** • **`M-g M-y`**	(**pel-goto-symbol-any-buffer**)	Prompt using for imenu symbol of all loaded menu supported buffers and move point to the selection. • Provide input completion using the currently selected method (emacs-default, ido, ivy or helm). • Select the default completion method by customization setting **pel-use-imenu-anywhere**. • Modify user interface currently used with **`M-g <f4> y`**. • The command sets a ref-marker before moving. Return to previous location by typing **`M-,`**