↖ Topic Index

Emacs Tab Bar

Operation	Keystroke	Function	Note
Emacs Tab Bar Emacs >= 27.1 Last updated on:	Emacs 27.1 introduced a tab bar that is quite useful to maintain an Emacs window "*context*". PEL key bindings for the management of tab bar are placed within the pel:window prefix, some are inside the PEL **pel-∑window/** window **hydra** in ∑ **Windows** 2026-02-01		Note: several commands an key bindings were introduced in Emacs 28 and later. They are not yet all documented here.🚧
Open this PDF file. See also: ∑ **Help/Info**	`<M-f11> M-= <f1>`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the ∑ **Tab Bar** 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.
∑ **Customize Emacs Tab Bar control**	`<M-f11> M-= <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs tab bar management support.
Toggle tab-bar-mode	`<M-f11> M-= M-=` `✱ <f7> w M-T`	(**tab-bar-mode** &optional ARG)	Toggle the tab bar in all frames. ☝ See **pel-∑window/** window **hydra** in ∑ **Windows** for Emacs >= 27.1
Create/delete tabs			
Do following command inside a new tab Emacs >= 28.1	• `C-x t t` • `<M-f11> M-= M-,`	(**other-tab-prefix**)	Display the buffer of the next command in a new tab. • The next buffer is the buffer displayed by the next command invoked immediately after this command (ignoring reading from the minibuffer). • Creates a new tab before displaying the buffer, or switches to the tab that already contains that buffer. • When '**switch-to-buffer-obey-display-actions**' is non-nil, '**switch-to-buffer**' commands are also supported.
Create a new tab	• `C-x t 2` • `<M-f11> M-= M-\` • `⌘-t`	• (**tab-bar-new-tab** &optional ARG FROM-NUMBER) • (**tab-new** &optional ARG FROM-NUMBER)	Create a new tab ARG positions to the right. • If a negative ARG, create a new tab ARG positions to the left. • If ARG is zero, create a new tab in place of the current tab. • If no ARG is specified, then add a new tab at the position specified by '**tab-bar-new-tab-to**'. • Argument addressing is relative in contrast to '**tab-bar-new-tab-to**', where argument addressing is absolute. • If FROM-NUMBER is a tab number, a new tab is created from that tab.
Duplicate tab retaining same window layout Emacs >= 28.1	• `C-x t n` • `<M-f11> M-= M-|` • `⌘-T`	• (**tab-bar-duplicate-tab** &optional ARG FROM-NUMBER) • (**tab-duplicate** &optional ARG FROM-NUMBER)	Clone the current tab to ARG positions to the right. • If a negative ARG, create a new tab ARG positions to the left. • If ARG is zero, create a new tab in place of the current tab. • If no ARG is specified, then add a new tab at the position specified by '**tab-bar-new-tab-to**'. • Argument addressing is relative in contrast to '**tab-bar-new-tab-to**', where argument addressing is absolute. • If FROM-NUMBER is a tab number, a new tab is created from that tab.

Tear window in tab bar
See: ∑ **Windows**
	`C-x w ^ t`
	• `<f11> w i t`
	`✱ <f7> T`
		(**tab-window-detach**)	Move the selected window to a new **tab**.
			• This command removes the selected window from the configuration stored on the current tab, and makes a new tab with that window in its configuration.

Create a tab for specified buffer	• `C-x t b` • `<M-f11> M-= M-b`	(**switch-to-buffer-other-tab** BUFFER-OR-NAME)	Prompt for a buffer and switch to buffer BUFFER-OR-NAME in another tab. • Like C-x 5 b (which see), but creates a new tab.
Visit a file in a new tab	• `C-x t f` • `C-x t C-f` • `<M-f11> M-= M-f`	(**find-file-other-tab** FILENAME &optional WILDCARDS)	Prompt for FILENAME. , edit file FILENAME, in another tab. • Like C-x 5 C-f (which see), but creates a new tab. • If WILDCARDS is non-nil, FILENAME can include widcards, and all matching files will be visited.
Open a dired buffer for specified directory	• `C-x t f` • `C-x t C-f` • `<M-f11> M-= M-d`	(**dired-other-tab** DIRNAME &optional SWITCHES)	"Edit" directory DIRNAME. Like 'dired' but make a new tab.
Close tab	• `C-x t 0` • `<M-f11> M-= M-c` • `⌘-w` `✱ <f7> w C-t`	(**tab-close** &optional TAB-NUMBER TO-NUMBER)	Close the tab specified by its absolute position TAB-NUMBER. • If no TAB-NUMBER is specified, then close the current tab and switch to the tab specified by 'tab-bar-close-tab-select'. • Interactively, TAB-NUMBER is the prefix numeric argument, and defaults to 1. • TAB-NUMBER counts from 1. • Optional TO-NUMBER could be specified to override the value of 'tab-bar-close-tab-select' programmatically with a position of an existing tab to select after closing the current tab. • TO-NUMBER counts from 1.
Close all other tabs	• `C-x t 1` • `<M-f11> M-= M-C`	(**tab-close-other** &optional TAB-NUMBER)	Close all tabs on the selected frame, except the tab TAB-NUMBER. TAB-NUMBER counts from 1 and defaults to the current tab (which happens interactively).
Close tab by name	`<M-f11> M-= M-k`	(**tab-bar-close-tab-by-name** NAME)	Prompt for the tab name and close it.
Restore most recently closed tab	`<M-f11> M-= M-u`	(**tab-undo**)	Restore the most recently closed tab.
Navigate across tabs	Move across (select) the tabs in the current Emacs frame with the following commands. • PEL supports shortcut keys for Emacs running in graphics mode on macOS using the `Fn` key which it configures as the Hyper modifier. This is not available when emacs is running in terminal mode.		
Select tab by number	• `⌘-1` • … … … • `⌘-9` • `M-1 <M-f11> M-t` • … … … • `M-9 <M-f11> M-t`	(**tab-select**)	• **On macOS** with Emacs running in graphical mode, PEL dynamically sets the **tab-bar-select-tab-modifiers** user-option variable to `(super)` , which effectively activates these quick ⌘ key bindings. This is similar to **macOS Safari tab keys**. • On other systems, and for Emacs running in terminal mode, use numeric prefix argument key(s) followed by `<M-f11> M-t`. This provides the extra flexibility of being able to select a larger range of tabs; you can compose an integer using numeric prefix argument keys.
Select next tab	• `<M-f11> M-= M-]` • `⌘-}` • `C-<tab>`	(**tab-next** &optional ARG)	Switch to next tab. • With n numeric prefix argument select next ARG tab
Select previous tab	• `C-x t 0` • `<M-f11> M-= M-[` • `⌘-{` • `C-S-<tab>`	(**tab-previous** &optional ARG)	
Select last tab	`<M-f11> M-= M-l`	(**tab-last** &optional ARG)	Switch to the last tab or ARGth tab from the end of the tab bar. • Interactively, ARG is the prefix numeric argument; it defaults to 1, which means the last tab on the tab bar. For example, `C-u 2` or `M-2` prefix selects the tab before the last tab.
Select recent tab	`<M-f11> M-= M-r`	(**tab-recent** &optional ARG)	Switch to ARGth most recently visited tab. • Interactively, ARG is the prefix numeric argument and defaults to 1.
Select tab by name • **Possibly create it**	• `C-x t RET` • `<M-f11> M-= M-s`	(**tab-switch** NAME)	Switch to the tab by NAME. • Default values are tab names sorted by recency, so you can use `M-n` to get the name of the most recently visited tab, the second most recent, and so on. • When the tab with that NAME doesn't exist, create a new tab and rename it to NAME.

Operation	Keystroke	Function	Note
Manage tabs			
Rename a tab	`<M-f11> M-= M-a`	(**tab-rename** NAME &optional TAB-NUMBER)	Give the tab specified by its absolute position TAB-NUMBER a new NAME. • If no TAB-NUMBER is specified, then rename the current tab. • Interactively, TAB-NUMBER is the prefix numeric argument, and defaults to the current tab. • TAB-NUMBER counts from 1. • Interactively, prompt for the new NAME. • If NAME is the empty string, then use the automatic name function '**tab-bar-tab-name-function**'.
Move tab to the right/left	• `C-x t m` • `<M-f11> M-= M-m`	(**tab-move** &optional ARG)	Move the current tab ARG positions to the right. • Interactively, ARG is the prefix numeric argument and defaults to 1. • If ARG is negative, move the current tab ARG positions to the left. • Argument addressing is relative in contrast to '**tab-bar-move-tab-to**', where argument addressing is absolute.
Restore tab previous window configuration	`<M-f11> M-= M-p`	(**tab-bar-history-back**)	Restore a previous window configuration used in the current tab. • This navigates back in the history of window configurations.
Cancel restoration of tab window layout	`<M-f11> M-= M-n`	(**tab-bar-history-forward**)	Cancel restoration of the previous window configuration. • This navigates forward in the history of window configurations.