↖ Topic Index

Scrolling

Description	Keystroke	Function	Note

Scrolling
○ GUI-Emacs scroll bars
• Recenter text in window
• Vertical scrolling
  • Smooth scrolling
  • Scroll full screen
  • Scroll lock mode
○ PEL window scrolling
  • hydra, vertical & multi-window vertical scroll
• Horizontal scrolling
• Auto-revert tail mode
○ Follow mode
 	
Last updated on:
	Emacs supports both ***vertical*** and ***horizontal*** scrolling, even when Emacs is running in text mode. In GUI mode Emacs has scroll bars you can use.
	
	Emacs has **several vertical** scroll modes and several global variables to control scrolling behaviour as described in the Textual Scrolling page.
	• Single window scroll, where the navigation commands for up and down are translated to scroll operations when scroll is possible.
	• Dual window scroll, implemented by the PEL package, where a PEL scroll command in one window is reflected in the other, associated window.
	• The scroll-all-mode, where all windows are scrolled together when one of the native Emacs scroll commands is used.
	• Emacs also has the **Follow Mode**, described at the bottom of this document. See PEL binding for follow-mode.
	PEL also provides 1-line scroll commands.
	Note that when Emacs is running in Terminal (TTY) mode, mouse scrolling is normally disabled.
	• PEL activates the ability to scroll with the mouse in terminal mode when the xterm-mouse-mode is activated (it can be enabled by **<f11><f12>**).
	
	See also: Σ **Mouse**
	
	2025-03-22		

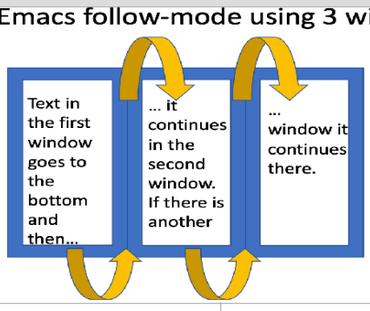
Open this PDF file. See also: Σ **Help/Info**	**<f11> | <f1>**	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the Σ **Scrolling** local PDF. If the prefix argument (like **C-u** or **M--**) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
Σ **Customize** PEL Scrolling control	**<f11> | <f2>**	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize PEL Scrolling support. • If OTHER-WINDOW is non-nil (use **C-u**), display in other window.
Σ **Customize** Emacs Scrolling control	**<f11> | <f3>**	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs Scrolling support groups: frame, follow, smooth-scrolling.
GUI Scroll bars	Emacs running in **graphical mode** supports vertical and horizontal scroll bars.		
Toggle Vertical scroll bar on all frames	**<f11> | B V**	(**scroll-bar-mode** &optional ARG)	Toggle vertical scroll bars on all frames (Scroll Bar mode). • This applies to all frames that exist, as well as new frames to be created in the future.
Toggle Vertical scroll bar on current frame	**<f11> | B v**	(**toggle-scroll-bar** ARG)	Toggle whether or not the selected frame has vertical scroll bars. • With ARG, turn vertical scroll bars on if and only if ARG is positive. The variable '**scroll-bar-mode**' (in the frame customize group) controls which side the scroll bars are on when they are turned on; if it is nil, they go on the left.
Toggle Horizontal scroll bar mode on all frames	**<f11> | B H**	(**horizontal-scroll-bar-mode** &optional ARG)	Toggle horizontal scroll bars on all frames for Emacs running **graphics mode**. • This command applies to all frames that exist and frames to be created in the future. • If the prefix argument is positive, enable the mode, zero or negative, disable the mode.
Toggle Horizontal scroll bar on current frame	**<f11> | B h**	(**toggle-horizontal-scroll-bar** ARG)	Toggle whether or not the selected frame has horizontal scroll bars. • With ARG, turn horizontal scroll bars on if and only if ARG is positive.
Recentering in current window	The following command is not a scrolling operation but can help reposition buffer. ☝ It can be quite useful to quickly position the line where the cursor is located to the centre, top or bottom of the window, effectively quickly '*scrolling*'.		
Position current line to window's Center / Bottom / Top . **Refresh screen.** See also: Σ **Windows**	• **C-l** • **<f11> C-l** • **<numkeypad 5>**	(**recenter-top-bottom** &optional ARG)	Without argument: moves the current line to window: center -> top -> bottom. • The **<5>** key on numeric keypad can also be used when available and active. • See Σ⌨**Numkeypad** for more information on how to control access to this. With arg: centre first: • **C-u C-l C-l C-l C-l** • ➜ center ➜ bottom ➜ center ➜ top With negative arg: bottom first: • **C-- C-l C-l C-l** • ➜ bottom ➜ center ➜ top With arg 0: top first: • **M-0 C-l C-l C-l** • ➜ top ➜ bottom ➜ center
	• With numeric positive: move current line to window top position N • With negative numeric: move current line to bottom window position: -1 := last line • PEL provides the **<f11> C-l** key binding because some modes use **C-l** as a prefix key.		
Vertical Scrolling	Emacs default vertical scrolling mechanism makes the screen jumps about the size of half a window when the cursor gets close to the top or the bottom of the window. This mode was originally designed to reduce processing requirements. See the following sections of the programming manual for info: • Textual Scrolling • Vertical Fractional Scrolling		
• **Smooth Scrolling**	If you find the behaviour of vertical scrolling annoying, you can activate a smooth scrolling. 📦 The smooth-scrolling package provides the **smooth-scrolling external package** that makes Emacs provide a much smoother scrolling experience. PEL activates smooth scrolling when the **pel-use-smooth-scolling** user option is set to **t**. Turn it on or off with the following command.		
Toggle smooth scrolling mode	**<f11> | s**	(**smooth-scrolling-mode** &optional ARG)	Toggle smooth scrolling mode on/off. If a numeric argument is used, a positive one turn smooth scrolling on while a negative one turns it off. 📦 Requires the **smooth-scrolling** package. Actvated by **pel-use-smooth-scrolling**
• **Scroll full screen**	Emacs provides the following standard commands to scroll text in the current window. These commands scroll without having to enable a special scroll mode (just as the PEL provided single line scroll commands below). However that also work in the single window scroll and the scroll all mode.		
Scroll up by near full screen	• **C-v** • **<PgDown>**	(**scroll-up-command** &optional ARG)	• Use an argument number (M-number) before the **C-v** to specify the number of lines to scroll. In **cua-mode**, **C-v** is used for paste. ☛ With PEL, if **pel-with-cua-paste** user option is set to t, **C-v** is bound to yank, otherwise it uses Emacs default (used for scroll).
Scroll down by near full screen	• **M-v** • **<PgUp>**	(**scroll-down-command** &optional ARG)	Use an argument number (M-number) before the **C-v** to specify the number of lines to scroll
• **Scroll lock mode**	Once enabled, scroll the window with the **<up>** and **<down>** keys. Also keys that normally move point by line or paragraph will scroll the buffer by the respective amount of lines instead and point will be kept vertically fixed relative to window boundaries during scrolling.		
• **1 Window Scroll**	The single window scroll mode allows using normal cursor movement keys to scroll the window.		
Toggle Scroll-Lock Mode	**<f11> | l**	(**scroll-lock-mode** &optional ARG)	Toggle single window scroll mode. A buffer-local minor mode for pager-like scrolling. • With prefix argument ARG, a positive argument enables it, a negative disables it.
• **All Windows Scroll**	In this mode **all** visible windows are scrolled together.		
Toggle Scroll All Mode	**<f11> | a**	(**scroll-all-mode** &optional ARG)	Toggle shared scrolling in same-frame windows (Scroll-All mode). • With prefix argument ARG, a positive argument enables it, a negative disables it.
	• When Scroll-All mode is enabled, scrolling commands invoked in one window apply to **all visible windows** in the same frame.		
• **Scroll other window**	The *other* window is the *next* window in the list of windows displayed in the Emacs frame. The following commands can be used to scroll the line of that other window.		
Scroll other window up by ~ 1 full window	• **Esc <PgDown>** • **C-M-v**	(**scroll-other-window** &optional ARG)	Scroll the text of the *other* window one near-full window down (toward the bottom of the buffer). • If a number argument is specified, that identifies the number of lines to scroll. • A negative argument inverses the direction.

Description	Keystroke	Function	Note
Scroll other window down by ~ 1 full window	• **Esc <PgUp>** • **C-M-S-v**	(**scroll-other-window-down** &optional LINES)	Scroll the text of the *other* window one near-full window up (toward the top of the buffer). • If a number argument is specified, that identifies the number of lines to scroll. • A negative argument inverses the direction.
• **PEL scroll** • supports multi-window sync of vertical scrolling • with 2 windows or more, as selected • in any major-mode • Hydra for scrolling: • vertical 1-line • horizontal 1-column	PEL provides a line scrolling mechanism that can include 2 or more windows that are scrolled in sync: the windows are part of a ***sync-scroll group***. • The first two commands, **pel-scroll-up** and **pel-scroll-down** scroll the text of current window, and *any window part of the sync-scroll group*, without moving the point. • Control the set of windows that are part of the *sync-scroll group* with the **pel-toggle-scroll-sync**, **pel-add-window-to-scroll-sync** and **pel-remove-window-from-scroll-sync** commands. The first one creates a set of 2 windows: the current one and its next window. • The windows part of the sync-scroll set will scroll together only when the **pel-scroll-up**, **pel-scroll-down** commands are used, but also when the following keys are used: **pel-home** (bound to **<home>**, and **pel-end** (bound to **<end>**). Nothing else unfortunately. See Σ **Navigation**. • While sync-scroll is active you can scroll the current window only with the commands **pel-scroll-down-only-this** and **pel-scroll-up-only-this**. • These commands will temporary disable the sync-scroll and will re-establish it right after. • 🚧 This is not a pure scroll mode: moving point inside a window will not scroll but if the window content is refreshed you may loose the lining up of your windows. ☝ Its behaviour is sufficient for several tasks, though, and permit the scrolling of a group of 2 or more windows, any one of them.	

PEL scroll **hydra** : **C-<f7>**	`scroll` | `Other`
	`C-<up>: up` | `M-?: hint cfg`
	`C-<down>: down` | `?: hint`
	`C-<left>: left` | `<C-f7>: cancel`
	`C-<right>: right`
		PEL scroll **hydra** provides keys to scroll vertically by 1 line and horizontally by 1 column. Start the by typing **C-<f7>** followed by any of the hydra keys. Then use any of the hydra keys by themselves. Close the hydra by typing **C-<f7>** again.
		📦 Requires the **hydra** external package. PEL activates when **pel-use-hydra** is **t.**	

Scroll up/down by N lines	For the next 2 command: repetition and inverse movement supported by numeric argument : Positive **N** identifies a repetition count. Negative **N** identifies repetition in the other direction.		
Scroll up 1 line	• **M-<down>** ✱ **<f7> M-<down>** ✱ **C-<f7> C-<down>**	(**pel-scroll-up** &optional N)	Move text 1 line up (same direction as forward) toward the end of buffer.
Scroll down 1 line	• **M-<up>** ✱ **<f7> M-<up>** ✱ **C-<f7> C-<up>**	(**pel-scroll-down** &optional N)	⚠ **M-<down>** and **M-<up>** do not work in **Ɱ Org-Mode**; these are bound to org-mode commands. However, with **hydra** installed, you can either use the Σ **Windows hydra** with its **<f7>** key prefix to to access versions of **M-<down>** and **M-<up>** that do work. The Σ **Windows hydra** has lots of features. You might prefer the simpler scroll **hydra** with its **C-<f7>** key prefix and then use **C-<down>** and **C-<up>**. Move text 1 line down (same direction as backwards), toward the top of buffer.
Scroll other window up by 1 line	**M-S-<f5>**	(**pel-scroll-up-other** &optional N)	Scroll the text of the *other* window one line down (toward the bottom of the buffer). • If a number argument N is specified, that identifies the number of lines to scroll. • A negative argument inverses the direction.
Scroll other window down by 1 line	**M-S-<f6>**	(**pel-scroll-down-other** &optional N)	Scroll the text of the *other* window one line up (toward the top of the buffer). • If a number argument N is specified, that identifies the number of lines to scroll. • A negative argument inverses the direction.
• **Sync-scroll group**	Use the following commands to establish the windows that are part of the sync-scroll group.		
• **Toggle PEL window scroll mode**	**<f11> | |**	(**pel-toggle-scroll-sync** &optional N)	Toggles the PEL window scroll-lock sync. If currently on, turn it off. ⚠ This command does not activate a minor-mode, therefore there's no indication. If currently off, place the current and next window inside the list of windows that must be scrolled together: the PEL window scroll sync group. • When turning it on, **you can identify the other window** with a a numeric argument. The value of **N** can be 2,4,6 or 8 and correspond to the cardinal points represented by the keypad cursor keys: 8 (up) 4 (left) 6 (right) 2(down)
• **Add current window to the PEL window scroll group**	**<f11> | +**	(**pel-add-window-to-scroll-sync**)	Add current window to the PEL scroll sync group. • If PEL scroll sync is currently off, it adds the current and next window to the PEL window scroll sync group.
• **Remove current window from the PEL window scroll group**	**<f11> | -**	(**pel-remove-window-from-scroll-sync**)	Remove current window from the PEL window scroll sync group. • If less than 2 windows remain in the group, the command display a warning. • It's possible to leave only 1 window inside the PEL window scroll sync group. That allows adding another window later. To turn it off completely re-issue to command or use the pel-toggle-scroll-sync command twice.
• **Scroll current window only when sync-scroll is active**	If you wish to **only scroll the current window while a sync-scroll group is active**, use the following 2 commands. • Use a prefix argument to scroll by several lines, or use the repeat command (bound to **<f5>** under PEL).		
• **Scroll up** **Bring text ahead into view.**	**<f11> | <down>**	(**pel-scroll-up-only-this** &optional N)	Move text N (defaults to 1) lines up (same direction as forward) toward the end of buffer. • Only scroll current window by N lines. No other windows part of the sync-scroll group scroll. The command temporary disable the sync-scroll and re-establish it after. • The command is only available during a sync-scroll group is active.
• **Scroll down** **Bring text behind into view.**	**<f11> | <up>**	(**pel-scroll-down-only-this** &optional N)	Move text N lines down (same direction as backwards), toward the top of buffer. • Only scroll current window by N lines. No other windows part of the sync-scroll group scroll. The command temporary disable the sync-scroll and re-establish it after. • The command is only available during a sync-scroll group is active.
Horizontal Scrolling for windows	Scrolling horizontally can be useful for very long lines or narrow windows. See Horizontal Scrolling • Left and right scrolling are also part of the PEL scroll Hydra: 📦 Requires the **hydra** external package. With PEL user option **pel-use-hydra** set to **t.** • To have the Hydra hint off when the Hydra activates set the **hydra-is-helpful** user option to nil (but then you can still toggle it on/off with **?**. ⚠ If the **C-<f7>** bindings do not work, use any of the **<f7>** window hydra to first load it and then the scroll hydra will work.		
Scroll window 1 column left	• **<f11> | }** • **<f11> | <right>** ✱ **C-<f7> C-<right>**	(**pel-scroll-left** &optional N)	Scroll the entire window left by 1 column. • A numeric argument N can be used to identify more columns
Scroll window 1 column right	• **<f11> | {** • **<f11> | <left>** ✱ **C-<f7> C-<left>**	(**pel-scroll-right** &optional N)	Scroll the entire window right by 1 column. • A numeric argument N can be used to identify more columns
Scroll window (lines) left	• **C-x <** • **C-<PgDown>**	(**scroll-left** &optional ARG SET-MINIMUM)	Scroll window left. • This command is disabled by default. The first time you use it it will prompt to enable it.
Scroll window (lines) right	• **C-x >** • **C-<PgUp>**	(**scroll-right** &optional ARG SET-MINIMUM)	Scroll window right. • This command is disabled by default. The first time you use it it will prompt to enable it.
Automatic scroll to end of buffer			
Toggle auto-revert tail mode See more in: Σ **File-mngt**	• **<f11> | t** • **<f11> f r t**	(**auto-revert-tail-mode** &optional ARG)	Toggle reverting tail of buffer when the file grows. • With a positive ARG enable Auto-Revert Tail Mode , a negative disables it. • When Auto-Revert Tail Mode is enabled, the tail of the file is constantly followed, as with the shell command 'tail -f'. This means that whenever the file grows on disk (presumably because some background process is appending to it from time to time), this is reflected in the current buffer. You can edit the buffer, turn this mode off and on again as you please. ⚠ Make sure the background process has stopped writing before saving the file!

Description	Keystroke	Function	Note

Follow Mode
 	
See also: **Windows**
	Emacs has a scroll all windows mode which applies all scroll commands to all visible windows. To support mouse wheel or scroll bar you need to implement extra code as suggested by the **Emacs Wiki Scroll All Mode** page.
	 
			When Emacs follow-mode is used on 2 or more windows, these windows show the text of the same buffer spread across these windows that act as a one continuous stream.
			
			• Follow mode is a minor mode that combines windows into one tall virtual window. This is accomplished by two main techniques:
			  • The windows always displays adjacent sections of the buffer. This means that whenever one window is moved, all the others will follow. (Hence the name Follow mode.)
			  • Should point (cursor) end up outside a window, another window displaying that point is selected, if possible. This makes it possible to walk between windows using normal cursor movement commands.
			• Follow mode comes to its prime when used on a large screen and two or more side-by-side windows are used. The user can, with the help of Follow mode, use these full-height windows as though they were one.

Toggle follow-mode See also: **Windows**	• `<f11> | f` • `<f11> w f`	(**follow-mode** &optional ARG)	Toggle Follow mode. With a prefix argument ARG, enable Follow mode if ARG is positive, and disable it otherwise.

Scrolling — Reference

Topic	Note
Scroll two opened buffers in a split window at the same time	Use **scroll-all-mode** I got it to work briefly (when (SL) was displayed in the mode line lighter) but have not been able to get it to work ever since. The scroll-lock-mode behaves like it is documenting in the source code; which is just to prioritize scrolling instead of moving point in the lines.
Emacs Wiki - Scroll All Mode	
Emacs Wiki - Scroll Lock	
GNU Emacs Manual - Scrolling	
GNU Emacs Manual - Auto Scroll	
GNU Emacs - Follow Mode	
Can I scroll the windows in emacs synchronously? @ StackOverflow	