## Windows — Managing Emacs Windows

The contract of the contract o	These basic facilities can be extended by several built-in and external packages:  windmove, built-in, activated by PEL, with different key bindings to preserve ability to shift-mark when moving across windmove, built-in, which provides the ability to restore previous window pane layouts. PEL activates it when perceivage in the windows perceivage in the windows in the perceivage in the windows in the perceivage in the windows in other frames.  Expendibar Seesions  Emacs Lisp Windows section  Page links:  Follow Mode  See more links beside the Hydra description some rows below.  Frames  Operations  Frames  Lisp Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows can be dedicated to specifi	s text with cursor.  I-use-winner user option is t.  s
The contract of the contract o	frames. These basic facilities can be extended by several built-in and external packages:  windmove, built-in, activated by PEL, with different key bindings to preserve ability to shift-mark when moving across windmove, built-in, activated by PEL, with different key bindings to preserve ability to shift-mark when moving across windmove, built-in, activated by PEL, with different key bindings to preserve ability to shift-mark when moving across windmove.  Excustomize  Key-Chords  Key-Chords  Frames  Speedbar  Speedbar  Speedbar  Speedbar  Speedbar  Speedbar  Windows can be dedicated to specific buffers, for example by Speedbar (see Speedbar).  Windows section  Windows with the same buffers can operate as a single flow with follow mode.  PEL provides extra commands and key bindings:  It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  Windows, the Menu key is mapped to the super prefix key (s-).  Windows, the Menu key is mapped to the hyper key. Below the is con is used to represent the Menu key under the xind with the xterm-mouse-mode enabled. With PEL, use <f11><f11><f11></f11></f11></f11></f11>	Il-use-winner user option is t. s
Some in the part of the part o	See also:  **Scustomize**  **Ekey-Chords*  **Erames*  **Espeedbar*  **Es	Il-use-winner user option is t. s
<ul> <li>Controllar Service Controllar Service</li></ul>	<ul> <li>See also:         <ul> <li>∑ Customize</li> <li>∑ Key-Chords</li> <li>∑ Frames</li> <li>∑ Speedbar</li> <li>∑ Speedbar</li> <li>∑ Scrolling</li> <li>∑ Sessions</li> <li>Emacs Lisp Windows section</li> </ul> </li> <li>Page links:         <ul> <li>Follow Mode</li> <li>See more links beside the Hydra description some rows below.</li> </ul> </li> <li>I ayout-restore</li></ul>	s
**Section Color   Price and Co	<ul> <li>■ Key-Chords</li> <li>■ Frames</li> <li>■ Speedbar</li> <li>■ Speedbar</li> <li>■ Scrolling</li> <li>■ Sessions</li> <li>■ Emacs Lisp</li> <li>Windows section</li> <li>Page links:</li> <li>■ Follow Mode</li> <li>■ See more links beside the Hydra description some rows below.</li> <li>■ Coperations on windows can be denoted to specific bendings and terminal mode only one frame is visible at a time though.</li> <li>■ Mace-window, we extends the C-x o command by displaying Ace target in the windows' upper left corner for quick per left per left corner for quick per left corner for quick per left corner for quick per left per l</li></ul>	ck navigation and access to buttons.
** Execution** So in contrast cash up pulsame-accesswindow uson or grow or it.  ** Security**  ** Windows can be dedicitated to specific fulfiller, for example by Secordar (per s. Specifillar)*  ** Windows can be dedicitated to specific fulfiller, for example by Secordar (per s. Specifillar)*  ** Security**  ** Security**  ** Security**  ** Secondary (per s. Specifillar)*  ** Secondary (per s. Speci	Enacs Lisp Windows section  Pel activates it when pel-use-ace-window user option is t.  * key-chord,  to activate dual-key chords to move across windows.  PEL activates it when pel-use-key-chord to windows section  Page links: Follow Mode See more links beside the Hydra description some ows below.  Present a section of the window service in the pel-use-ace-window user option is t.  * key-chord, to activate dual-key chords to move across windows.  PEL provides extra commands to dedicate window to specific purposes to activated by pel-use-window-purpose.  * Several windows with the same buffers can operate as a single flow with follow mode.  PEL provides extra commands and key bindings:  It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  * On macOS, in graphics mode only, the % key is mapped to the super prefix key (s-).  * On Windows, the Menu key is mapped to the hyper key. Below the ★ icon is used to represent the Menu key used to the super prefix key (s-).  * They can also be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <f11><f12>  Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or in terminal mode only one frame is visible at a time though.</f12></f11></f11>	se user-option.
Securities	<ul> <li>Espeedbar         <ul> <li>Speedbar</li> <li>Scrolling</li> <li>Sessions</li> </ul> </li> <li>Emacs Lisp         Windows section         <ul> <li>Windows section</li> </ul> </li> <li>Page links:         <ul> <li>Follow Mode</li> <li>See more links</li> <li>Teside the Hydra description some ows below.</li> <li>Lisp Seedbar (see Speedbar)</li> <li>Windows to activate dual-key chords to move across windows.</li> <li>PEL provides extra commands and key bindings:</li></ul></li></ul>	se user-option.
Securious	<ul> <li>Scrolling</li> <li>Sessions</li> <li>Emacs Lisp</li> <li>Windows section</li> <li>Page links:</li> <li>Follow Mode</li> <li>See more links</li> <li>See more links</li> <li>To activate dual-key chords to move across windows.</li> <li>Windows section</li> <li>Windows section</li> <li>PEL provides extra commands and key bindings:</li> <li>It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.</f11></li> <li>It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.</f11></li> <li>In graphics mode, mouse operations are available.</li> <li>In graphics mode, mouse operations are available.</li> <li>They can also be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <f11><f12></f12></f11></li> <li>Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or in terminal mode only one frame is visible at a time though.</li> </ul>	se user-option.
**Section** **Emants laps** **Windows action** **Page links: **Chook Mode ** **Chook Mode ** **See in the links action in the links a	* Windows can be dedicated to specific buffers, for example by Speedbar (see ★ Speedbar ).  * windows section  * window-purpose can be use to dedicate window to specific purposes a citivated by pel-use-window-purpose.  * Several windows with the same buffers can operate as a single flow with follow mode.  * PEL provides extra commands and key bindings:  * It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  * On macOS, in graphics mode only, the ★ key is mapped to the super prefix key (s-).  * On Windows, the Menu key is mapped to the hyper key. Below the ❖ icon is used to represent the Menu key used to represen</f11>	
**Mindows section  **Windows sec	* window-purpose can be use to dedicate window to specific purposes a activated by pel-use-window-purpose.  * Several windows with the same buffers can operate as a single flow with follow mode.  * PEL provides extra commands and key bindings:  * It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  * On macOS, in graphics mode only, the % key is mapped to the super prefix key (s-).  * On Windows, the Menu key is mapped to the hyper key. Below the ❖ icon is used to represent the Menu key understand the provides extra commands and key bindings:  * It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  * On Windows, the Menu key is mapped to the hyper key. Below the ❖ icon is used to represent the Menu key understand the provides extra commands and key bindings:  * It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  * On Windows, the Menu key is mapped to the hyper key. Below the ❖ icon is used to represent the Menu key understand the provides extra commands and key bindings:  * It adds several key bindings:  * On macOS, in graphics mode only in graphics and terminal modes.  * On windows can be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <f11><f12> the provides extra commands and key bindings:  * It adds several key bi</f12></f11></f11></f11></f11>	
Pick provided soft a commands and key bindings:	PEL provides extra commands and key bindings:  • It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  • See more links beside the Hydra description some lows below.  PEL provides extra commands and key bindings:  • It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.  • On Windows, the Menu key is mapped to the super prefix key (s-).  • On Windows, the Menu key is mapped to the hyper key. Below the short is used to represent the Menu key is in graphics mode, mouse operations are available.  • They can also be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <f11><f12>  • Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or in terminal mode only one frame is visible at a time though.</f12></f11></f11></f11>	ıder Windows.
** It acks several key brundings under the cft1s key prife. There are available in both graphics and terminal modes.  ** It is acks several key brundings under the cft1s key prife. There are available in both graphics and terminal modes.  ** It is present the price of the price	<ul> <li>It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.</f11></li> <li>Follow Mode</li> <li>See more links beside the Hydra description some ows below.</li> <li>It adds several key bindings under the <f11> key prefix. These are available in both graphics and terminal modes.</f11></li> <li>On windows, the Menu key is mapped to the super prefix key (s-).</li> <li>On Windows, the Menu key is mapped to the hyper key. Below the ❖ icon is used to represent the Menu key used to re</li></ul>	der Windows.
© Commodition in Membrane and any of the pill by a proposed to the paper parks key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the proposed to the paper park key (a).  © Commodition in the park (a).  © Commodition	• In graphics mode only, the # key is mapped to the super prefix key (s-).  • On Windows, the Menu key is mapped to the hyper key. Below the is con is used to represent the Menu key useside the Hydra description some ows below.  • In graphics mode, mouse operations are available.  • They can also be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <f11><f12>  Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or in terminal mode only one frame is visible at a time though.</f12></f11>	der Windows.
• See on window into sees the Hybrid Robot Sees the Hybrid R	<ul> <li>See more links beside the Hydra description some ows below.</li> <li>Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or ir In terminal mode only one frame is visible at a time though.</li> </ul>	ider Windows.
** "Inguigation mode, modes operations are available.  ** "They can also be resided in terminal mode, with the stem-mouse mode enabled. With PEIL, use <pre><pre> ** "They can also be resided in terminal mode, with the stem-mouse mode enabled. With PEIL, use <pre> ** "They can also be presided to windows in other frames, whether traces is morning in apprition mode on in terminal mode. With the stem-mouse mode with the stem-mouse mode on the stem of the stem o</pre></pre></pre>	<ul> <li>In graphics mode, mouse operations are available.</li> <li>They can also be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <f11><f112></f112></f11></li> <li>Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or ir In terminal mode only one frame is visible at a time though.</li> </ul>	
Containing PEL	description some rows below.  They can also be enabled in terminal mode, with the xterm-mouse-mode enabled. With PEL, use <### <### color of the properties of the pell of the	
Countering Period or in terminal mode or in	Ows below.  Operations on windows can be applied to windows in other frames, whether Emacs is running in graphics mode or in In terminal mode only one frame is visible at a time though.	to toggle the xterm-mouse-mode.
Questimate PDF Fig.		terminal mode.
Customize PEL	Description DDF (in the control of t	
Customize PEL	upen this Pur tile. □ <f11> w <f1> □ (pel-help-pdf &amp;optional OPEN- □ Open the ∑ Windows local PDF. If the prefix argume</f1></f11>	nt (like <b>C-u</b> or <b>M</b> ) is used, then it oper
Coustomize PEL Window control  Coustomize Finance (11) w < f3>		, , , , , , , , , , , , , , , , , , , ,
inflow control  Customize Emacs Window upper groups: window, ace-window. See-window. See-window see-window. See-window ace-window. See-window see-window. See-window see-window. See window. See wind	the other way around.	
inflow control  Customize Emacs Window upper groups: window, ace-window. See-window. See-window see-window. See-window ace-window. See-window see-window. See-window see-window. See window. See wind	© Customize PEL	
window control  OTHER-WINDOW)  window control  window window info  see:  -		n other window.
window control  OTHER-WINDOW)  window indicates the second of the secon	(cal customing Funces Windows are under which the control of the c	
A windrassize does not uses its own group. It places its customization inside the Emacs convenience group instead. PEL opens that profit is look of Windrassize user options them Show window info see.  **C+11> 7 d w **C+11> 7 d		ace-window, ace-window-display,
Stow window Info Stow Information about window, information about window, information about window in Info Info Store Information Information about window Info Information about window Information about the Information		tite customization inside the Emacs
Show information about window, information about window, information about window, information among depends on command argument.  **CFID** I w #  **CFID** I		
Sec: 4.13. ? d w (\$1.00 cm) (\$1.0		· · · · · · · · · · · · · · · · · · ·
### A propriet of the plant is commally our can only see the bottom of the info. See in "Massage" buffer with M-O or C-19 prefix print displayer control variables in a *pel-window-info* buffer same parameters with M-O or M-19 prefix print displayer control variables in a *pel-window-info* buffer has button that open help on the variable providing access to outstandow-info buffer has button that open help on the variable providing access to outstandow-info buffer has button that open help on the variable providing access to outstandow-info buffer has button that open help on the variable providing access to outstandow denomination buffer.  ### With a co-window-display-mode user-option on, the window number is shown on the left of the mode-line.  ### With a co-window-display-mode user-option on, the window number is shown on the left of the mode-line.  ### With a co-window-display-mode user-option on, the window number is a control of the mode-line.  ### With a co-window-display-mode user-option on, the window number is a control of the mode-line.  ### With a co-window-display-mode user-option on, the window number is a control of the mode-line.  ### With a co-window-display-mode user-option on, the window number of each window inside the left hand side of its mode line.  ### With a control of the mode-line.  ### With a control of the mode-l	W. 1	
Wink #-0 or C-u prefix: print display-buffer control variables in a "pel-window-info" buffer buffer same as M-0 but appends to the buffer. Les to collect into on several windows.	Demustifying Emoco	
With Mar 1: same as M-D but appends to the buffer. Use to collect info on several windows, window frame parameters window frame parameters window frame parameters window frame parameters.      With ace-window-display-mode user-option on, the window number is shown on the left of the mode-line.      **Type £11> <22> o ace-window-display-mode to open the customize buffer to change it.      Activating this will increase your Emacs in it time. Instead, use ace-window-display-mode, a minor mode that displays the ace window number of ace-window display-mode.      **C£11> w. If ace-window-display-mode to open the customize buffer to change it.      **Activating this will increase your Emacs in it time. Instead, use ace-window-display-mode, a minor mode that displays the ace window number of each window instead the left hand side of its mode line.      **C£11> w. If ace-window-display-mode ace-window-display-mode ace-window display-mode. The ace-window display-mode ace-window display-mode ace-window display-mode ace-window display-mode ace-window display-mode. The ace-window display-mode ace-window display-mode ace-window display-mode ace-window display-mode.      **C£11> w. If ace-window-display-mode ace-window-display-mode ace-window display-mode ace-window display-mode ace-window display-mode ace-window display-mode.      **C£11> w. If ace-window-display-mode ace-window-display-mode ace-window display-mode ace-window display-mode.      **C£11> w. If ace-window-display-mode ace-window-display-mode ace-window display-mode ace-window.      **Type £11> x. Extra to the second window-display-mode ace-window-display-mode ace-window display-mode ace-window-display-mode.      **C£11> w. If ace-window-display-mode	Window Manager	_
The "pel-window-info" buffer has button that open help on the variable providing access to customization buffer.		•
Example   Customization buffer   Customiza	Frame parameters	
ace-window # on		neip on the variable providing access to
**Type £f11> £f22 o ace-window-display-mode to open the customize buffer to change it.  **Activating this will increase your Emacs init time. Instead, use ace-window-display-mode, £f11> w # , to activate it manually.  **Toggie showing ace-window fished with time for the customize buffer to change it.  **Activating this will increase your Emacs init time. Instead, use ace-window-display-mode, £f11> w # , to activate it & ace-window number of sex window inside the left hand side of its mode line.  **Toggie showing ace-window display-mode & period window inside the left hand side of its mode line.  **Toggie the ace-window external package. **EL use pel-use-ace-window.**  **PEL Window hydra external package. **EPL user option pel-use-hydra set to 1 activate it & create a hydra to speed up navigation and management of window.  **To the follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the £f7> key again.  **While active the Hydra Hith its shown in the minibuffer (as shown below. You can hit several different in succession without having to type the £f7> key again.  **While active the Hydra Hith its shown in the minibuffer (as shown below. Type the ? key to toggie the hit find off to back on.  **To have the Hydra hith the £f7> key, shown below. Type the ? key to toggie the hit find off to back on.  **You can prefet these commands with prefix agruments such as C-u and numerical prefix with H-O, H-I H-I H-9 to commands that accept them.  **The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-Yund/ and ends with the same name as the content.  **Elia zertical/**  **The name of the PEL window hydra commands are not listed below. They all have a name that begins with the file of the window management hydra hith menu shows up in the minibuffer area as soon as one of its keys is pressed:  **Specific to window hydra commands are not listed below. They all have a name that begins with pel-Yund/ a	Customization burier.	
Toggle showing ace-window # on the command of the command for the command f	ace-window # on	
Coggle showing ace-window # on window may be continued with the con	▼ Mode Line Type <f11> <f2> o ace-window-display-mode to open the customize buffer to change it.</f2></f11>	
ace-window # on window mode line  PEL Window Hydra Quickly:  Navigate through window mode line  Needs hydra external package.  PEL user option pel-use-hydra set to 1 activate it & create a Hydra to speed up navigation and management of windows. To start this hydra, hit the ₹₹₹ New again.  Then follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  Then follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  Then follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different his cuscession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different his cuscession without having to type the filt in foll of the \$₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different his succession without having to type the \$₹ Po key again.  The follow by typing the PEL Window hydra calculates set the hydra-is-helpful user option to nil (but then you can still toggle it on/off with ₹ Po keys provides a partially overlapping feature set but has a different his with \$₹ Po keys provides a partially overlapping feature set but has a different his his with \$₹ Po keys provides a partially overlapping feature set but has	⚠ Activating this will increase your Emacs init time. Instead, use ace-window-display-mode, <f11> w # , to activativativativativativativativativativa</f11>	te it manually.
ace-window # on window mode line  PEL Window Hydra Quickly:  Navigate through window mode line  Needs hydra external package.  PEL user option pel-use-hydra set to 1 activate it & create a Hydra to speed up navigation and management of windows. To start this hydra, hit the ₹₹₹ New again.  Then follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  Then follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  Then follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different in succession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different his cuscession without having to type the ₹₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different his cuscession without having to type the filt in foll of the \$₹ Po key again.  The follow by typing the PEL Window Hydra keys, shown below. You can hit several different his succession without having to type the \$₹ Po key again.  The follow by typing the PEL Window hydra calculates set the hydra-is-helpful user option to nil (but then you can still toggle it on/off with ₹ Po keys provides a partially overlapping feature set but has a different his with \$₹ Po keys provides a partially overlapping feature set but has a different his his with \$₹ Po keys provides a partially overlapping feature set but has	Toggle showing	that diaplays the acc window number of
PEL Window Hydra Quickly.  Nevaga windows.  Navagate through windows.  Navaga window.  Navaga wi		
PEL Window Hydra Ouickly:  Navigate through windows.  Navigate through window windows.  Navigate through window windows.  Navigate through windows.  Navigate through window window.  Navigate through window.  Navigate		PEL use pel-use-ace-window.
Hydra Quickly.  Navigate through windows.  Navi		•
<ul> <li>Nexigate through windows</li> <li>Swap windows</li> <li>Swap windows</li> <li>Open buffer in different window</li> <li>Open buffer in different window</li> <li>Open buffer in different window</li> <li>Close windows</li> <li>Open buffer in different window</li> <li>Close windows</li> <li>Close windows</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on.</li> <li>While active the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hydra feet by the yound and when the Hydra buffer (as shown below). Type the ? by to and B to change the buffer currently visible in the current window.</li> <li>Use the q key to quit from buffers that can be dismissed like the 'Helpfu Lies by to and B to change the buffer currently visible in the current window.</li> <li>The name of the PEL window buffer active such as C-u and numerical prefix with M-0, M-1 M-9 to command the tom the H</li></ul>	PEL Window Needs hydra external package. DEL user option pel-use-hydra set to t activate it & create a Hydra to speed up	navigation and management of windows
- Swap windows - Open buffer in different window - Close window - To have the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on 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 ? - Volume 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 ? - Volume 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 ? - Volume 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 ? - Volume the Hydra Hint is shown in the minibuffer (as shown below). Type the ? key to toggle the hint info off or back on The when the Hydra Hint is shown in the minibuffer (as shown below). They all have a can be dismissed like the "Help' buffer. Use b and B to change the buffer currently visible in the current window.  Substitute q key to quit from buffers that can be dismissed like the "Help' buffer. Use b and B to change the buffer currently visible in the current window.  The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wmd/ and ends with the same name as the command function listed in the Function column. For example, pel-∑wmd/windowo-up is bound to <f1? <="" up=""> - The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wmd/ and ends with the same name as the content.  Split P split Window line are not listed below. They all have a name that begins with pel-∑wmd/ and ends with the same name as the service of the window undown that accept them.  The purpose of the window management hydra hint menu shows up in the minibuffer are as soon as one of its keys is pressed.  Split P split Window line on window line on window line on window line on w</f1?>		· · · · · · · · · · · · · · · · · · ·
• 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 ?.  So Doen buffer in different window Close window Kill buffer Create/Spilt normal/side/root windows Kill buffer Create/Spilt normal/side/root windows Resize window Fit isze to buffer content Fit pertical/ borizontal layout Window layout Mindow Mindow layout Mindow Mindow layout Mindow Min		
Open buffer in different window clifferent window close window likill buffer contact split normal/side/root windows. Will buffer contact split normal/side/root windows. Besize window command bunder of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wind/and ends with the same name as the command function listed in the Function column. For example, pel-∑wind/windowo-up is bound to <f?> Split F split W layout history command layout sistory windows window w</f?>		
different window Close window Kill buffed Create/Split Content/Split Content/Split Content Filip vertical Normal/Stode/root Normal/Stode/root windows Fit size to buffer Content Clip vertical Normal/Stode/root Normal/Stode/root Normal/Stode/root Normal/Stode/root Normal/Stode/root Windows Fit size to buffer Content Clip vertical Normal/Stode/root Norma	• Open huffer in	
Kill buffer   Create/Split   Northous   The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  The ace-window command function listed in the Function column. For example, pel_Yound/windowov-up is bound to cf7> <up> The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel_Yound/ and ends with the same name as the command function listed in the Function column. For example, pel_Yound/windowov-up is bound to cf7&gt; <up> The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel_Yound/ and ends with the same name as the command function listed in the Function column. For example, pel_Yound/windowov-up is bound to cf7&gt; <up> The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel_Yound/ and ends with the same name as the command function listed in the Function column. For example, pel_Yound/windowov-up is bound to cf7&gt; <up> The name of the PEL window management hydra hint menu shows up in the minibuffer area as soon as one of its keys is pressed:  Split Fr  Split Window in the Function column. For example, pel_Yound/windowov-up is bound to cf7&gt; <up> Split Fr  Split Window info  Split Fr  Split Window info on window info</up></up></up></up></up>	different window of the command by name with M-x	
Create/Split normal/side/root windows command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  **The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  **The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  **The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wind/and ends with the same name as the command function listed in the Function column. For example, pel-∑wind/windomove-up is bound to <f1> <ul> <li>**The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wind/and ends with the same name as the command function listed in the Function column. For example, pel-∑wind/windomove-up is bound to <f1> <ul> <li>**The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wind/and ends with the same name as the command function listed in the Function column. For example, pel-∑wind/windomove-up is bound to <f1> <ul></ul></f1></li></ul></f1></li></ul></f1>		ently visible in the current window.
**The ace-window command bound to C-x o key provides a partially overlapping feature set but has a different key assignment than the Hydra # key.  **The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-∑wnd/ and ends with the same name as the command function listed in the Function column. For example, pel-∑wnd/windnove-up is bound to <fr> **Elize to buffer content** - Elip vertical/ horizontal layout   More window and agenet hydra hint menu shows up in the minibuffer area as soon as one of its keys is pressed:  **Split** - Split** - Spl</fr>		commands that accept them.
### The name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-\( \) window command function listed in the Function column. For example, pel-\( \) window info on window purpose external package, window help and to change the dedication and purpose of window buffer.  **Split frame**  **Split Window**  **Split Frame**  **Split Window**  **Split Frame**  **Split Window**  **Split Window**  **Split Frame**  **Split Window**  **Split Window**  **Split Frame**  **Split Window**  **Split Frame**  **Split Window**  **Split Window**  **In name of the PEL window hydra commands are not listed below. They all have a name that begins with pel-\( \) window of \$\( \) to \$\( \) \(		assignment than the Hydra # key.
A snapshot of the window management hydra hint menu shows up in the minibuffer area as soon as one of its keys is pressed:    SplitF		
content  Filip vertical/ horizontal layout  Window layout history  Dedicate windows Didicate windows Didicate windows Didicate window Layout Follow mode  Resize  Close  O: this R: kill buf/vin  At root		
Flip vertical/ horizontal layout Window layout history Dedicate windows Didicate window purpose Flolow mode Recenter buffer  Split frame Split Window Spurpose Follow mode Recenter buffer  Split frame Split Window Spurpose Another hydra is available to display window help and to change the dedication and purpose of window and/or its associated buffer.  The purpose specific commands require the window-purpose external package, Window information in the minibuffer  Activate that hydra with <ff>For the first 2 columns perform an action and display window information in the minibuffer  The keys in the "Dedicate" column toggle the state of window, buffer purpose Window, buffer purpose Window purpose  Another hydra with <ff>For the M-p key toggle the purpose-mode. The M-p and M-w  Split frame Split window Signature  All keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose and wind</ff></ff>	content	
Morizontal layout   Window layout   Window layout   Window layout   More point   Si continuous   More point	Delici De	uffer Other
history Dedicate windows Didicate windows Didicate windows Didicate window Durpose Follow mode Recenter buffer  Recenter buffer  Recenter buffer  Another hydra is available to display window information in the minibuffer.  Activate that hydra with <ff? "dedicate"="" 2="" action="" all="" an="" and="" buffer="" burpose="" c-s-<up="" column="" columns="" dedication.="" display="" first="" in="" information="" key="" keys="" m-b="" m-i="" m-p="" m-w="" minibuffer.="" of="" perform="" purpose="" purpose-mode.="" state="" the="" tinfo="" toggle="" window="" window,="">: shorter   1: others   b: next buffer   f: follow-mod   F: prev buffer   F: prev</ff?>	horizontal layout / 8: root 1 2: - s: fix size <up>: 1 =: balance 0: this K</up>	
Dedicate windows Didicate window purpose window window window purpose window Did window win	/ At root C Comp. O no last layout   Clofts C   we shorter   1. others   h	
Durpose Follow mode Recenter buffer  Split frame Split Window Change Layout Switch to the pel-Sbuffer Hydra by typing <f7><f7><f7> Another hydra is available to display window help and to change the dedication and purpose of window window information in the minibuffer.  Activate that hydra with <f7> All keys in the first 2 columns perform an action and display window, buffer purpose and window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  Nove Point By win # Resize window Close window Close window Close window Close window Will buffer, Change buffer. Recenter Recente</f7></f7></f7></f7>	Dedicate windows / 6: root C- <down>: I state layout   v: shorter   v:</down>	: prev buffer I: info
Purpose Follow mode Recenter buffer  Split frame Split Window Change Layout Switch to the pel-∑buffer Hydra by typing <ff><ff><ff><ff><ff><ff><ff><ff><ff><ff< td=""><td></td><td></td></ff<></ff></ff></ff></ff></ff></ff></ff></ff></ff>		
Split frame   Split Window   Info on window   Change Layout   Switch to the   Split frame   Split Window   Change Layout   Switch to the   Split frame   Split Window   Switch to the   Switch to the   Split frame   Split Window   Switch to the   Split frame   Split frame   Switch to the   Switch to the   Split frame   Switch to the	The state of the s	· · · · · · · · · · · · · · · · · · ·
Change Layout Switch to the pel-∑buffer Hydra by typing <f7><f7><f9> See ∑ Buffers  Another hydra is available to display window help and to change the dedication and purpose of window and/or its associated buffer.  The purpose specific commands prequire the window-purpose external package, activated by pel-use-window-purpose user-option.  • Activate that hydra with <f7> M-i M-i. • All keys in the first 2 columns perform an action and display window information in the minibuffer. • The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication. • The M-p key toggle the purpose-mode. The M-b and M-w    M-i</f7></f9></f7></f7>	Follow mode \ 6: side \	<f7>: cancel</f7>
Switch to the pel-∑buffer Hydra by typing <f7><f9> See ∑ Buffers Recenter Help, quit, cancel  Another hydra is available to display window help and to change the dedication and purpose of window and/or its associated buffer.  The purpose specific commands prequire the window-purpose external package, activated by pel-use-window-purpose user-option.  • Activate that hydra with <f7> M-i M-i.  • All keys in the first 2 columns perform an action and display window information in the minibuffer.  • The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  • The M-p key toggle the purpose-mode. The M-b and M-w  Switch to the plane is available to display window and/or its associated buffer.  Bedicate Window info Other  M-d: window M-i: current M-p: toggle purpose mode vipose in the window purpose dedication.  **M-d: window purpose in the window</f7></f9></f7>	Recenter nutter	
Another <a href="hydra">hydra</a> is available to display window help and to change the dedication and purpose of window and/or its associated buffer.  The purpose specific commands require the <a href="window-purpose">window-purpose</a> external package, activated by <a href="pel-use-window-purpose">pel-use-window-purpose</a> user-option.  Activate that <a href="hydra">hydra</a> with <a href="hydra">hydra</a> with <a href="hydra">first 2</a> columns perform an action and display window info minibuffer.  All keys in the first 2 columns perform an action and display window infommation in the minibuffer.  The keys in the "Dedicate" column toggle the state of window purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  A-d: window window purpose window and/or its associated buffer.  Window info  Other  M-d: window  M-i: current  M-p: toggle purpose mode  Cup:  The M-p key toggle the purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  Window info  Other  Current  M-p: toggle purpose mode  Cup:  Change  Countries  Coun	Split traine Split willdow into oil willdow into explicit nesize willdow close willdow ki	
The purpose specific commands require the window-purpose external package, activated by pel-use-window-purpose user-option.  Activate that hydra with <f7> M-i M-i.  All keys in the first 2 columns perform an action and display window information in the minibuffer.  The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  The M-p key toggle the purpose-mode. The M-b and M-w  Activated by pel-use-window-purpose user-option.  Dedicate  Window info  M-i: current  M-p: toggle purpose mode  Y: hint  M-w: window purpose  All keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w</f7>	Change Layout By win #	Help, quit, cancel
Activate that hydra with <f7> M-i M-i.  All keys in the first 2 columns perform an action and display window infomration in the minibuffer.  The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  Dedicate  Window info  Other  M-d: window  M-i: current  M-p: toggle purpose mode  ?: hint  M-w: window purpose    All keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  Activate that hydra with <f7> M-i.  M-d: window info  Other </f7></f7>	<u>Change Layout</u> By win #	
All keys in the first 2 columns perform an action and display window information in the minibuffer.  The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  All keys in the first 2 columns perform an action and display window information in the minibuffer.  M-d: window  M-i: current  Column:  W-p: toggle purpose mode  Column:  Column:  M-w: window purpose  Column:  Colu	Switch to the pel-∑buffer Hydra by typing <f7><f9> See ∑ Buffers Split Window Nove Folia Nesize Window Nove Folia Nesize</f9></f7>	ciated buffer.
All vindow information in the minibuffer.  The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  The M-p key toggle the purpose-mode. The M-b and M-w  All vindow M-d: window M-d: window M-d: window M-d: window M-b: buffer purpose  (up>: 1	Split Window Change Layout By win # CI  Switch to the pel-∑buffer Hydra by typing <f7><f2> See ∑ Buffers Research Buffers By win down and/or its association and purpose of window and/or its association and window and/or its association and window and/or its association and window and/or i</f2></f7>	
window information in the minibuffer.  • The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  • The M-p key toggle the purpose-mode. The M-b and M-w  window  M-i: current  M-p: toggle purpose mode cup::    N-p: toggle purpose mode cup::   N-p: toggle purpose cup::   N-p: toggle	Switch to the pel-∑buffer Hydra by typing <f7><f1>  Switch to the pel-∑buffer Hydra by typing <f7><f2><f3>  See ∑ Buffers  Another hydra is available to display window help and to change the dedication and purpose of window and/or its assortance. The purpose specific commands prequire the window-purpose external package, was activated by pel-use-window.</f3></f2></f7></f1></f7>	-purpose user-option.
<ul> <li>The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.</li> <li>The M-p key toggle the purpose-mode. The M-b and M-w</li> </ul> <ul> <li>M-b: buffer purpose   <up>: 1</up></li> <li>Adown:   <down: li=""  <=""> <li>Adown:   </li> <li>Ado</li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></down:></li></ul>	Change Layout Switch to the pel-∑buffer Hydra by typing <f7><f7><f9> See ∑ Buffers  Another hydra is available to display window help and to change the dedication and purpose of window and/or its assort The purpose specific commands prequire the window-purpose external package, activated by pel-use-window  • Activate that hydra with <f7> M-i M-i.  Dedicate  Window info</f7></f9></f7></f7>	-purpose user-option.
• The M-p key toggle the purpose-mode. The M-b and M-w	Change Layout Switch to the pel-∑buffer Hydra by typing <f7><f7><f9> See ∑ Buffers  Another hydra is available to display window help and to change the dedication and purpose of window and/or its assortance.  The purpose specific commands prequire the window-purpose external package, activated by pel-use-window.  Activate that hydra with <f7> M-i M-i.  All keys in the first 2 columns perform an action and display</f7></f9></f7></f7>	-purpose user-option.  Other
	Switch to the pel-∑buffer Hydra by typing <fr> Another hydra is available to display window help and to change the dedication and purpose of window and/or its assort The purpose specific commands prequire the window-purpose external package, display activated by pel-use-window information in the minibuffer.  Activate that hydra with <fr> All keys in the first 2 columns perform an action and display window information in the minibuffer.  The keys in the "Dedicate" column toggle the state of  Activate that hydra with <fr> M-d: window window information in the minibuffer.  M-d: window window information in the minibuffer.  M-b: buffer purpose continuous indicates window information in the minibuffer.  M-b: buffer purpose continuous indicates window information in the minibuffer.</fr></fr></fr>	-purpose user-option.  Other  M-p: toggle purpose mode  it is hint
keys activate purpose-mode if it is not already active.	Change Layout  Switch to the pel-∑buffer Hydra by typing <f7><f7><f9>  Another hydra is available to display window help and to change the dedication and purpose of window and/or its assored The purpose specific commands require the window-purpose external package, activated by pel-use-window  • Activate that hydra with <f7> M-i M-i.  • All keys in the first 2 columns perform an action and display window information in the minibuffer.  • The keys in the "Dedicate" column toggle the state of window, buffer purpose and window purpose dedication.  M-b: buffer purpose dedication:  M-w: window purpose   Close window   M-is current   M-b: buffer purpose   Close window   M-is current   M-is</f7></f9></f7></f7>	-purpose user-option.  Other  M-p: toggle purpose mode  it is hint

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Move point to identified window			ursor key bindings described below. In some circumstances, these key bindings can conflict an be translated to Meta-cursor keys that are bound to Org-mode operations.
Esc-cursor keys	PEL provides the following	ng user options to control the key bir	dings:
for windmove	<ul> <li>This affects the behavio</li> <li>1 0 Several Linux disthis, otherwise don't becommands.</li> </ul>	our of the <esc> cursor key bindings tros map <b>C-M-</b> bindings such as <b>C-</b></esc>	is on by default on macOS and Windows, but off on Linux. in org buffer as well to ensure a regular navigation across all buffers.  M- <right> and C-M-<left> If this is not the case for your Linux system, you can activate the Esc C- bindings in replacement for the C-M- bindings you need to access several Emacs on by default.</left></right>
Move to window above	<pre>   <f11> <up>   <f1> <up>   <f1> <up>   <esc> <up>     %-<up>     %-<up>     *-<up>     * <f7> <up>   </up></f7></up></up></up></up></esc></up></f1></up></f1></up></f11></pre>	(windmove-up &optional ARG)	Select the window above the current one.  • With no prefix argument, or with prefix argument equal to zero, "up" is relative to the position of point in the window; otherwise it is relative to the left edge (for positive ARG) or the right edge (for negative ARG) of the current window.  • If no window is at the desired location, an error is signalled.    With PEL, the yu key-chord is also available when key-chord is available and active.    See   Key-Chords.
Move to window below	• <f11> <down> • <f1> <down> • <f1> <down> • <esc> <down> • %-<down> • *-<down> • *-<down> • *-<down> • bn</down></down></down></down></down></esc></down></f1></down></f1></down></f11>	(windmove-down &optional ARG)	Select the window below the current one.  • With no prefix argument, or with prefix argument equal to zero, "down" is relative to the position of point in the window; otherwise it is relative to the left edge (for positive ARG) or the right edge (for negative ARG) of the current window.  • If no window is at the desired location, an error is signalled.    With PEL, the bn key-chord is also available when key-chord is available and active.  See   Key-Chords.
Move to window at left	<pre> • <f11> <left> • <f1> <down> • <esc> <left> • %-<left> • \$-<left>  * <f7> <left> • gf</left></f7></left></left></left></esc></down></f1></left></f11></pre>	(windmove-left &optional ARG)	Select the window to the left of the current one.  • With no prefix argument, or with prefix argument equal to zero, "left" is relative to the position of point in the window; otherwise it is relative to the top edge (for positive ARG) or the bottom edge (for negative ARG) of the current window.  • If no window is at the desired location, an error is signalled.    With PEL, the gf   key-chord is also available when key-chord is available and active.    See     Key-Chords
Move to window at right	<pre>• <f11> <right> • <f1> <right> • <esc> <right> • %=<right> • \$-<right>  * <f7> <right>  ik</right></f7></right></right></right></esc></right></f1></right></f11></pre>	(windmove-right &optional ARG)	Select the window to the right of the current one.  • With no prefix argument, or with prefix argument equal to zero, "right" is relative to the position of point in the window; otherwise it is relative to the top edge (for positive ARG) or the bottom edge (for negative ARG) of the current window.  • If no window is at the desired location, an error is signalled.  •••••••••••••••••••••••••••••••••••
Move point to other window - C-u: swap - C-u C-u: delete	• C-x o * <f7> #</f7>	(other-window COUNT &optional ALL-FRAMES)	Select (move point) to other window. Select another window in cyclic ordering of windows.  With prefix argument consider all frames.  This is Emacs default behaviour for this key. And PEL's default: pel-use-ace-window = nil. Change it to activate the functionality described in next row.
Move to other window     Move to specified window Ace target	With only 2 windows in th.	(ace-window ARG) e current frame, move to the other w	Move to (and possibly operate on) window selected by an Ace target code.  Requires the <u>ace-window</u> external package. PEL downloads, installs and activates it when the <u>pel-use-ace-window</u> user option is set to t.
Operate on specified window	<ul> <li>With 3 windows or more: display an Ace target in the windows' upper left corner that identifies the window target:</li> <li>Type the displayed window number to move to that window. With C-x o you can also type one of the extra character before the window number:</li> </ul>		
See also: ∑ Customize	• x - delete windo		<ul> <li>m - swap windows</li> <li>c - copy window</li> </ul>
Demo: <u>C'est la Z, video 5</u>	<ul><li>v - split window</li><li>F - split window</li></ul>	r in the other window	<ul> <li>n - aw-flip-window: switch to the window previously used</li> <li>e - execute command other window</li> <li>b - split window horizontally</li> <li>o - maximize current window (delete others)</li> </ul>
	<ul> <li>This supports selecting windows in other frames (both in graphics and terminal mode)</li> <li>In graphics mode the other Emacs frames are in other OS window.</li> <li>In text terminal mode, other Emacs frames are hidden (as they occupy the exact same OS window): just one Emacs frame is displayed.</li> <li>An argument can be used to perform more operations:</li> <li>To force a window number prompt, use any negative prefix (including just typing C alone). Useful with several frames when current frame has 1 or 2 windows active.</li> <li>Prefixed with one C-u, does a swap between the selected window and the current window, so that the selected buffer moves to current window (and current buffer moves to selected window). The PEL <f11> w x key does the same (but does not prompt when there are only 2 windows.)</f11></li> <li>Prefixed with two C-u's, deletes the window identified by the window number.</li> </ul>		indow.  ey occupy the exact same OS window): just one Emacs frame is displayed.  including just typing C alone). Useful with several frames when current frame has 1 or 2  window and the current window, so that the selected buffer moves to current window (and  w x key does the same (but does not prompt when there are only 2 windows.)
Move point to next window • can specify all frames	<f11> w o</f11>	(pel-other-window &optional ALL-FRAMES)	Move to other window, like the original other-window.  • With any prefix argument consider all frames. Without argument move only within current frame.  • Useful when 'other-window' has been remapped to something like 'ace-window' and want to see where the <i>next</i> window is.
Move point to previous window can specify all frames	<f11> w 0</f11>	(pel-other-window-backward &optional N)	Select Nth previous window. n defaults to 1: meaning direct previous window.  with negative n: move as (abs n) but consider all frames. If n is positive consider only current frame.  This is the inverse of what does the standard (other-window).  This command might be useful when ace-window is not used.
Swap (eXchange) windows	• <f11> w x * <f7> x</f7></f11>	(ace-swap-windows)	Swap buffers of the current window with another. If 3 windows or more, a single digit shows up in the top-left corner identifying the number to type to swap to this window.  Requires the <a href="mailto:ace-window">ace-window</a> external package. PEL downloads, install and activates it when the <a href="mailto:pel-use-ace-window">pel-use-ace-window</a> user options is set to t.
Open Buffer in another window		buffer name is using the input comp	de another window. One command select (move point to) that window. The other does not. eletion method currently active (default, Ido, Helm,)
Display buffer in other window, don't select the other window.	• C-x 4 C-o • <f11> w b</f11>	(ido-display-buffer)  — — — — — — — — — — — — — — — — — — —	Display a buffer in other window but don't select it.
Select buffer in other window	• C-x 4 b • <f11> w B</f11>	(ido-switch-buffer-other-window)	Select buffer bufname in another window (switch-to-buffer-other-window). See Select Buffer.

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Close Windows	The following commands are	used to remove (close) windows. The	ne last row correspond to a set of four PEL commands bound to cursor keys.
Close this windows	• C-x 0 * <f7> 0</f7>	(delete-window &optional WINDOW)	This just closes the window and moves the cursor to the next window.
Close other (next) window	• <f11> w w * <f7> o</f7></f11>	(pel-close-other-window)	Close the other window. Hide its buffer, does not kill it.  Useful to close temporary window, like the help window, without having to move into it.
Close all other windows	• C-x 1 * <f7> 1</f7>	(delete-other-windows &optional WINDOW)	Maximize current window: make current window fill its frame. Close all other windows.
Close window identified by number	<f11> w k</f11>	(ace-delete-window)	Delete a window selected by a number, a number shown in the top-left corner of the window.  • If there's only 2 windows, kills the other window. If only 1 window is used, does not kill it.  • Needs <u>ace-window</u> external package.  • PEL downloads, installs and activates it when the <u>pel-use-ace-window</u> user options is set to t.
Maximize window identified by number	<f11> w m</f11>	(ace-maximize-window)  (ace-delete-other-windows)	Maximize specified window. Close all windows except the window selected by number, a number shown in the top-left corner of the window.  Needs <u>ace-window</u> external package. The old versions used ace-window-maximize, but newer versions use ace-delete-maximize-windows. PEL uses the one that is available. PEL downloads, install and activates it when the pel-use-ace-window user options is set to t.
Close a window identified by cursor direction	• ESC C-S- <right> • ESC C-S-<left> • ESC C-S-<down> • ESC C-S-<up> • <f1> C-S-<right> • <f1> C-S-<left> • <f1> C-S-<left> • <f1> C-S-<down> • <f1> C-S-<down> • <f1> C-S-<down> • <f1> C-S-<up> • <f11> C-S-<wip> • <f11> C-S-<right> • <f11> C-S-<ip> • <f11> C-S-<ip> • <f11> C-S-<left> • <f11> C-S-<down> • <f11> C-S-<down> • <f11> C-S-<up> * <f7> C-S-<right> * <f7> C-S-<left> * <f7> C-S-<down> * <f7> C-S-<down> * <f7> C-S-<up></up></f7></down></f7></down></f7></left></f7></left></f7></left></f7></left></f7></left></f7></right></f7></up></f11></down></f11></down></f11></left></f11></ip></f11></ip></f11></right></f11></wip></f11></up></f1></down></f1></down></f1></down></f1></left></f1></left></f1></right></f1></up></down></left></right>	pel-close-window-right)     (pel-close-window-left)     (pel-close-window-down)     (pel-close-window-up)	<ul> <li>Kill window pointed by the cursor's direction.</li> <li>The 4 different commands and shown in the same cell for convenience, one for each of the available cursors: <right>, <left>, <down> and <up>.</up></down></left></right></li> <li>There are 4 possible sets of bindings:</li> <li>3 sets of stand-alone commands:</li> <li>Commands with <f11> prefix, always available.</f11></li> <li>Commands with ESC prefix, available when pel-windmove-on-esc-cursor user option is on (set to t).</li> <li>Commands with <f1> prefix, available when pel-windmove-on-f1-cursor user option is on (set to t).</f1></li> <li>The Hydra-based commands, with the Hydra activated with any of the key sequences that use the <f7> prefix. Available when pel-use-hydra user option is set to t.</f7></li> </ul>
Close all windows showing buffer	• C-x w 0 • <f11> w 0</f11>	(delete-windows-on &optional BUFFER-OR-NAME FRAME)	Prompts for buffer name and delete all windows showing that buffer.  With M-0 prefix: delete only windows in the current terminal's frames. Any other prefix argument means that only windows in the current frame will be deleted.
Kill current buffer and close window See also:   Buffers	• C-x 4 0 * <f7> K</f7>	(kill-buffer-and-window)	Kill the current buffer and delete the selected window.
Kill current buffer	* <f7> k</f7>	(pel-kill-current-buffer)	Kill current buffer and close window without prompting unless it is modified. In Hydra only.
Create Window by splitting current window	The split-window-keep-	point user option controls whether p minimize redisplay. Change tempora	rrent one. The last row correspond to a set of four PEL commands bound to cursor keys.  oint is kept at the same vertical position in both windows (t, the default). If nil, Emacs adjust rily with: <f11> <f4> w s. Change permanently with: <f11> w <f3> 1 to access the</f3></f11></f4></f11>
Toggle split window point behaviour	<f11> w <f4> s</f4></f11>	(pel-toggle-split-window-keep-point)	Toggle the value of <b>split-window-keep-point</b> between values described above. Print description of new value. Change only affects current Emacs session, not stored.
Create new window below	• C-x 2 * <f7> 2</f7>	(split-window-below &optional SIZE)	Split current window into 2 windows. Leave point in top window. Same buffer in both.     Optional SIZE numerical argument identify line count of top window (if positive) or bottom window (if negative).
Create new window at right	• C-x 3 * <£7> 3	(split-window-right &optional SIZE)	Split current window into two side-by-side windows. Leave point in the left window. Same buffer in both.  Optional SIZE numerical argument identify column count of left-hand window (if positive) or right-hand window (if negative).
Create window at cursor direction	• ESC C- <right> • ESC C-<left> • ESC C-<down> • ESC C-<down> • Color C- • C-<right> • C-<left> • C-<left> • C-<left> • C-<down> • Cf1&gt; C-<down> • Cf1&gt; C-<down> • Cf1&gt; C-<up> • Cf11&gt; C-<up> • Cf11&gt; C-<left> • Cf11&gt; C-<up> * Cf7&gt; C-<up> * C-<up> * Cf7&gt; C-<up></up></up></up></up></up></up></up></up></up></up></up></left></left></left></left></left></up></up></down></down></down></left></left></left></right></down></down></left></right>	(pel-create-window-right & optional SIZE)     (pel-create-window-left & optional SIZE)     (pel-create-window-down & optional SIZE)     (pel-create-window-up & optional SIZE)	Create a window at the location pointed by the cursor's direction, and move point inside the new window.  Optional SIZE numerical argument identify either:  line count of top window (if positive) or bottom window (if negative).  column count of left-hand window (if positive) or right-hand window (if negative).  The 4 different commands and shown in the same cell for convenience, one for each of the available cursors: ⟨right⟩, ⟨left⟩, ⟨down⟩ and ⟨up⟩.  There are 4 possible sets of bindings:  3 sets of stand-alone commands:  Commands with ⟨f11⟩ prefix, always available.  Commands with ESC prefix, always available when pel-windmove-on-esc-cursor user option is on (set to t).  Commands with ⟨f1⟩ prefix, available when pel-windmove-on-f1-cursor user option is on (set to t).  The Hydra-based commands, with the Hydra activated with any of the key sequences that use the ⟨f7⟩ prefix. Available when pel-use-hydra user option is set to t.
Create Side Windows			ndows positioned at any of the four sides of a frame's <i>root</i> window.  ns the entire frame width under several vertically split windows.
Create new side window that holds current buffer.	• <f11> w \ 8 • <f11> w \ 2 • <f11> w \ 6 • <f11> w \ 4 * <f7> \ 8 * <f7> \ 2 * <f7> \ 6 * <f7> \ 4</f7></f7></f7></f7></f11></f11></f11></f11>	(pel-buff-in-side-win-top &optional N)     (pel-buff-in-side-win-bottom &optional N)     (pel-buff-in-side-win-right &optional N)     (pel-buff-in-side-win-left &optional N)	Place current buffer in a new, dedicated side window.  • By default the side window is at the bottom of the current frame.  • Use a numeric argument to specify a different side:  For N= 2, 4, 6 or 8, select window pointed by what is pointed by cursor positioned at the layout of numeric keypad:  8 := 'top 4 := 'left 6 := 'right 2 := 'bottom
Toggle display of side windows in the frame	• C-x w s • <f11> w M-s</f11>	(window-toggle-side-windows &optional FRAME)	Toggle display of side windows on current frame.  • If FRAME has at least one side window, delete all side windows on FRAME after saving FRAME's state in the FRAME's 'window-state' frame parameter. Otherwise, restore any side windows recorded in FRAME's 'window-state' parameter, leaving FRAME's main window alone. Signal an error if FRAME has no side windows and no saved state for it is found.
Create Frame Root Windows	·		re width or height of the frame, regardless of how many windows already exist in the frame.  und later only. On earlier versions of Emacs PEL implements the commands.
Split root window above	• <f11> w / 8 * <f7> / 8</f7></f11>	(pel-split-root-window-top &optional SIZE)	Split root window of current frame in two.  The current window configuration is retained in the lower window, the top window takes up the whole width of the frame.  Optional SIZE numerical argument sets line count of top window (if positive) or bottom window (if negative).

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Split root window below	C-x w 2 • <f11> w / 2 * <f7> / 2</f7></f11>	(split-root-window-below &optional SIZE) (pel-split-root-window-bottom &optional SIZE)	Split root window of current frame in two.  The current window configuration is retained in the top window, the lower window takes up the whole width of the frame.  Optional SIZE numerical argument sets line count of top window (if positive) or bottom window (if negative).
Split root window right	C-x w 3 • <f11> w / 6 * <f7> / 6</f7></f11>	(split-root-window-right &optional SIZE) (pel-split-root-window-right &optional SIZE)	Split root window of current frame into two side-by-side windows.  The current window configuration is retained within the left window, and a new window is created on the right, taking up the whole height of the frame.  Optional SIZE numerical argument identify column count of left-hand window (if positive) or right-hand window (if negative).
Split root window left	• <f11> w / 4 * <f7> / 4</f7></f11>	(Pel-split-root-window-left &optional SIZE)	Split root window of current frame into two side-by-side windows.  The current window configuration is retained within the right window, and a new window is created on the left, taking up the whole height of the frame.  Optional SIZE numerical argument identify column count of left-hand window (if positive) or right-hand window (if negative).
Resize Window Quickly with windresize	Requires the windresize	external package. 🔁 PEL activates	id (mapped to <f11> w r by PEL). it when pel-use-windresize user-option is set to t. v Hydra is active, taking over Hydra keys. Complete and return to Hydra with RET</f11>
Resize Window interactively	<f11> w r</f11>	(windresize &optional INCREMENT)	Resize windows interactively using the following minor mode keys.  • Use <b>RET</b> to complete or <b>C-g</b> to abort. Both exit the mode.
Resize window using cursors	<pre>     <right>     <left>     <down>     <up> </up></down></left></right></pre>	(windresize-right & optional N LEFT-BORDER FIXED-WIDTH)     (windresize-left & optional N LEFT-BORDER FIXED-WIDTH)     (windresize-down & optional N LEFT-BORDER FIXED-WIDTH)     (windresize-up & optional N LEFT-BORDER FIXED-WIDTH)	Resize the current window in the direction of the used cursor.  N is the number of lines by which moving borders.
Resize windows using direction opposite to cursor	• C- <right> • C-<left> • C-<down> • C-<up></up></down></left></right>	(windresize-right-minus)     (windresize-left-minus)     (windresize-down-minus)     (windresize-up-minus)	Same as the above commands but use the direction opposite to the cursor.
Resize window bottom-right	/	(windresize-bottom-right)	Call 'windresize-right' and 'windresize-down' successively.  In move-borders method, move the bottom-right edge of the window outwards.  In resize-window method, enlarge the window horizontally and shrink it vertically.
Resize window top- right	\	(windresize-up-right)	Call 'windresize-right' and 'windresize-up' successively.  In move-borders method, move the upper-right edge of the window outwards.  In resize-window method, enlarge the window both horizontally and horizontally.
Resize window top- left	M-/	(windresize-up-left)	Call 'windresize-left' and 'windresize-up' successively.  In move-borders method, move the upper-left edge of the window outwards.  In resize-window method, shrink the window horizontally and enlarge it vertically.
Resize window bottom-left	M-\	(windresize-bottom-left)	Call 'windresize-left' and 'windresize-up' successively.  In move-borders method, move the bottom-left edge of the window outwards.  In resize-window method, shrink the window both horizontally and vertically.
Reposition window	• C-M- <right> • C-M-<left> • C-M-<down> • C-M-<up></up></down></left></right>	(windresize-right-fixed)     (windresize-left-fixed)     (windresize-down-fixed)     (windresize-up-fixed)	Move the window to the direction identified by the cursor, keeping its width (or height) constant.
Set window resize/ reposition increment step	i	(windresize-set-increment &optional N)	Set the window resize increment step value to N.  • Use a numeric argument prefix to set N interactively:  • For example: M-4 i sets the increment to 4.
Increase the resize/ reposition increment step	+	(windresize-increase-increment &optional SILENT)	Increase the increment.  • If SILENT is non-nil, don't output a message.
Decrease the resize/reposition increment step	-	(windresize-decrease-increment &optional SILENT)	Decrease the increment.  If SILENT is non-nil, don't output a message.
Negate resize/ reposition increment	~	(windresize-negate-increment &optional SILENT)	Negate the increment value. Changes the direction of window resize operations.  • If SILENT is non-nil, don't output a message.
Balance Windows	• = • C-x +	(windresize-balance-windows)	Balance window sizes.
Delete current window	• 0 • C-x 0	(delete-window &optional WINDOW)	Delete current window  During my testing C-x 0 behaved like windresize-other-window instead.   Should investigate. 0 works fine though.
Delete other windows	• 1 • C-x 1	(windresize-delete-other- windows)	Delete other windows.
Split window vertically	• 2 • C-x 2	(windresize-split-window- vertically)	Split window vertically. Creates 2 windows: one on top of the other.
Split window horizontally	• 3 • C-x 3	(windresize-split-window- horizontally)	Split window horizontally. Creates 2 windows side by side.
Save window configuration	s	(windresize-save-window-configuration)	Save the current window configuration in the ring.
Restore window configuration	r	(windresize-restore-window-configuration)	Restore the previous window configuration in the ring.
Move point to other adjacent window	• M-S- <right> • M-S-<left> • M-S-<down> • M-S-<up></up></down></left></right>	(windresize-select-right &optional ARG)     (windresize-select-left &optional ARG)     (windresize-select-down &optional ARG)     (windresize-select-up &optional ARG)	Select the window identified by the cursor.  If ARG is nil or zero, select the window relatively to the point position.  If ARG is positive, select relatively to the top edge and select relatively to the bottom edge otherwise.
Move point to other window	o	(windresize-other-window)	Select other window.
Move point to previous window	p	(windresize-previous-window)	Select the previous window.
Move point to next window	n	(windresize-next-window)	Select other window.

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Set window layout and exit windresize	• x • RET	(windresize-exit)	Keep this window configuration and exit 'windresize'.
Cancel window layout and exit windresize	• q	(windresize-cancel-and-quit)	Cancel window resizing and quit 'windresize'.  • Restore window layout used before the entry into windresize mode.  • The layouts, are, however still available via winner-undo <f11> w p, with PEL.</f11>
Resize Window Using the base Emacs commands	The following commands are used to change the current window size. Except when used inside the hydra, none of these commands are easy to re-type quickly.  • The best way to use them is to type them once and then use a repeat key:  • Emacs native repeat key is C-x z once and then repeat more by only typing 'z'. PEL also binds the <f5> key to repeat.  • PEL also provides the Window Hydra (described above) which can be started with one of the following commands using the <f7> prefix. Once the Hydra is entered, commands can be issued again without any prefix.  Each of the first 5 commands below have 5 possible bindings:  • The Emacs default key binding using the C-x prefix.  • The commands with the default PEL <f11> prefix, always available.  • The commands with ESC prefix, available when pel-windmove-on-esc-cursor user option is on (set to t).  • The Hydra-based commands, activated with any of the key sequences that use the <f7> prefix. Available when pel-use-hydra user option is set to t.</f7></f11></f7></f5>		
Toggle fixed size window constraint	• <f11> w s s * <f7> s</f7></f11>	(pel-toggle-window-size-fixed &optional STRICT)	Toggle the fix size window constraint.  • With optional argument STRICT, this sets the 'window-size-fixed' variable which imposes a strict size constraint, preventing Emacs from changing the size of the window even if it would be necessary to, for example, display the mini buffer.  • By default, with no argument, the size restriction is not strict; it prevents most operations to change the window size but Emacs can still change the size if it must, for example, make place for the mini buffer.
Grow window taller	• C-x ^ • <f11> w s V • ESC M-<up> • <f1> M-<up> * <f7> V</f7></up></f1></up></f11>	(enlarge-window DELTA &optional HORIZONTAL)	Grow window taller by DELTA lines (defaults to 1), specify more with <b>C-u</b> n (or <b>M-</b> n) argument prefix.  • See note above for availability of various bindings.
Shrink window smaller	• <f11> w s v • ESC M-<down> • <f1> M-<down> * <f7> v</f7></down></f1></down></f11>	(shrink-window DELTA &optional HORIZONTAL)	Shrink height of window by DELTA lines (defaults to 1), specify more with <b>C-u</b> n (or <b>M-</b> n) argument prefix.  • See note above for availability of various bindings.
Grow windows wider	• C-x } • <f11> w s H • ESC M-<right> • <f1> M-<right> * <f7> H</f7></right></f1></right></f11>	(enlarge-window-horizontally DELTA)	Enlarge the current window horizontally.     See note above for availability of various bindings.
Shrink window narrower	• C-x { • <f11> w s h • ESC M-<left> • <f1> M-<left> * <f7> h</f7></left></f1></left></f11>	(shrink-window-horizontally DELTA)	Reduce the width of the current window.  See note above for availability of various bindings.
Make all windows the same size	• C-x + • <f11> w s = • ESC <kp-5> • <f1> <kp-5> * <f7> =</f7></kp-5></f1></kp-5></f11>	(balance-windows & optional WINDOW-OR-FRAME)	Balance the sizes of windows of WINDOW-OR-FRAME.  WINDOW-OR-FRAME is optional and defaults to the selected frame.  If WINDOW-OR-FRAME denotes a frame, balance the sizes of all windows of that frame. If WINDOW-OR-FRAME denotes a window, recursively balance the sizes of all child windows of that window.  See note above for availability of various bindings.
Reduce current window size if buffer is smaller than window	• C-x - • <f11> w s - * <f7> -</f7></f11>	(shrink-window-if-larger-than- buffer &optional WINDOW)	Shrink height of current window if its buffer doesn't need so many lines.  More precisely, shrink window vertically to be as small as possible, while still showing the full contents of its buffer.  Do not shrink window to less than 'window-min-height' lines. Do nothing if the buffer contains more lines than the present window height, or if some of the window's contents are scrolled out of view, or if shrinking this window would also shrink another window, or if the window is the only window of its frame.
Fit window size to current buffer's content	• C-x w - • <f11> w s . * <f7> .</f7></f11>	(fit-window-to-buffer &optional WINDOW MAX-HEIGHT MIN-HEIGHT MAX-WIDTH MIN-WIDTH PRESERVE-SIZE)	<ul> <li>Adjust size of WINDOW to display its buffer's contents exactly.</li> <li>WINDOW must be a live window and defaults to the selected one.</li> <li>If WINDOW is part of a vertical combination, adjust WINDOW's height. The new height is calculated from the actual height of the accessible portion of its buffer. The optional argument MAX-HEIGHT specifies a maximum height and defaults to the height of WINDOW's frame. The optional argument MIN-HEIGHT specifies a minimum height and defaults to 'window-min-height'. Both MAX-HEIGHT and MIN-HEIGHT are specified in lines and include mode and header line and a bottom divider, if any.</li> <li>If WINDOW is part of a horizontal combination and the value of the option 'fit-window-to-buffer-horizontally' is non-nil, adjust WINDOW's width. The new width of WINDOW is calculated from the maximum length of its buffer's lines that follow the current start position of WINDOW. The optional argument MAX-WIDTH specifies a maximum width and defaults to the width of WINDOW's frame. The optional argument MIN-WIDTH specifies a minimum width and defaults to 'window-min-width'. Both MAX-WIDTH and MIN-WIDTH are specified in columns and include fringes, margins, a scrollbar and a vertical divider, if any.</li> </ul>
Quick Window Layout Change	The following commands flip the layout of 2 windows: the current and <i>next</i> window between 2 horizontal windows to 2 vertical windows and vice versa.		
Flip 2 horizontal windows to 2 vertical ones	• <f11> w v * <f7> M-v</f7></f11>	(pel-2-vertical-windows)	Convert 2 horizontal windows into 2 vertical windows.  Flip the orientation of the current window and its next one.  The next window is placed at the right of the current window.
Flip 2 vertical windows to 2 horizontal ones	• <f11> w h * <f7> M-h</f7></f11>	(pel-2-horizontal-windows)	Convert 2 horizontal windows into 2 horizontal windows.  Flip the orientation of the current window and its next one.  The next window is placed below the current one.
Window Layout History	The following commands allow you to restore a previously used window layout. Two packages are available.  The <u>winner</u> package, a package that is part of the standard Emacs. PEL activates them when <b>pel-use-winner</b> user option is <b>t</b> .		
Restore an earlier window configuration	• C-c <left> • <f11> w p  * <f7> p</f7></f11></left>	(winner-undo)	Switch back to an earlier window configuration saved by Winner mode. In other words, "undo" changes in window configuration.
Restore a more recent window configuration	• C-c <right> • <f11> w n * <f7> n</f7></f11></right>	(winner-redo)	Restore a more recent window configuration saved by Winner mode.
Save/Restore window layout	The external layout-restorm. This needs investigation		el-use-restore-layout user-option set to t. This associates layouts to buffers.
Save Window layout	<f11> w 1 s</f11>	(layout-save-current)	Save the current layout, add a list of current layout to layout-configuration-alist.
Restore Layout	<f11> w 1 r</f11>	(layout-restore &optional BUFFER)	Restore the layout related to the buffer BUFFER, if there is such a layout saved in 'layout-configuration-alist', and update the layout if necessary.
Delete Layout	<f11> w 1 d</f11>	(layout-delete-current &optional BUFFER)	Delete the layout information from 'layout-configuration-alist' if there is an element list related to BUFFER.

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Dedicated Windows			ay that future windows operations do not affect the dedicated windows.  ute with the following command. Use <f11> w ? to show the current window state.</f11>
Toggle dedicated status of current window	• <f11> w d * <f7> M-i d</f7></f11>	(pel-toggle-window-dedicated)	Toggle the dedicated status of the current window, changing a normal window into a dedicated one and a dedicated window into a normal one.  ! Use with care after learning about dedicated windows.
Dedicate Window Purpose	The following commands are Some of them are also availa	available.	purposes activated by <b>pel-use-window-purpose</b> user-option.  nich is invoked with its <f7> M-i hydra head. The hydra invoked command also prints window on the following in the first substitution of th</f7>
Toggle purpose- mode	• <f11> w P P  * <f7> M-i M-p</f7></f11>	(purpose-mode &optional ARG)	Toggle Purpose mode on or off. This is a global minor mode.  • PEL window info <a href="https://www.ncbe.ncbe.ncbe.ncbe.ncbe.ncbe.ncbe.ncbe&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Show purpose-&lt;br&gt;mode control&lt;br&gt;options&lt;/th&gt;&lt;th&gt;&lt;f11&gt; w P ?&lt;/th&gt;&lt;th&gt;(pel-show-window-purpose-info&lt;br&gt;&amp;optional APPEND)&lt;/th&gt;&lt;th&gt;Show 'purpose-mode' control user-options in *pel-window-info* buffer.  • With non-nil optional APPEND argument; append text to the buffer.  • Provides quick access to help/customization buffer to define the various purposes.  After modifying these you must compile the settings. Use &lt;f11&gt; w P C&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Activate purpose-&lt;br&gt;mode settings&lt;/th&gt;&lt;th&gt;&lt;f11&gt; w P C&lt;/th&gt;&lt;th&gt;(pel-compile-window-purpose-&lt;br&gt;user-options)&lt;/th&gt;&lt;th&gt;Activate the latest window-purpose user-options.  • Always execute after modifying purpose-mode user-options&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Toggle window dedication to its current buffer&lt;/th&gt;&lt;th&gt;• &lt;f11&gt; w P B  * &lt;f7&gt; M-i M-b&lt;/th&gt;&lt;th&gt;(purpose-toggle-window-buffer-&lt;br&gt;dedicated &amp;optional WINDOW)&lt;/th&gt;&lt;th&gt;Toggle window WINDOW's dedication to its current buffer on or off.  The PEL Window Info &lt;a href=" https:="" th="" www.nys.gov.n<=""></a>
Toggle window	• <f11> w P W</f11>	(purpose-toggle-window-	Toggle window WINDOW's dedication to its purpose on or off.
dedication to its purpose	* <f7> M-i M-w</f7>	purpose-dedicated &optional WINDOW)	The PEL Window Info <u>hydra</u> instance of the command also print window info.
Close all non dedicated to their purpose or buffer	<f11> w P 1</f11>	(purpose-delete-non-dedicated-windows)	Delete all windows that aren't dedicated to their purpose or buffer.
Load a purpose- aware window layout	<fli><fli>W P L L</fli></fli>	(purpose-load-window-layout &optional NAME LAYOUT-DIRS)	Load a window layout. Prompt the user for the name of a window layout.  • It searches the layout in the default specified by defaults to 'purpose-layout-dirs'.  • If 'purpose-use-built-in-layouts', then 'purposebuilt-in-layouts-dir' is also searched.  • See 'purpose-find-window-layout' for more details.
Load a purpose- aware window layout from file	<f11> w P L 1</f11>	(purpose-load-window-layout- file &optional FILENAME)	Load window layout from file FILENAME, providing the default.
Save a purpose- aware window layout	<f11> w P L S</f11>	(purpose-save-window-layout NAME DIRECTORY)	Save a window layout. Prompt for NAME, the name to give the window layout, then prompt for DIRECTORY, the directory in which to save the layout. Tab completion provides default directory.
Save a purpose- aware window layout to file	<f11> w P L s</f11>	(purpose-save-window-layout- file &optional FILENAME)	Save window layout of current frame to file FILENAME.  If FILENAME is nil, use 'purpose-default-layout-file' instead.
Load most recent purpose-aware window layout	<f11> w P L r</f11>	(purpose-reset-window-layout)	Load most recent window layout from 'purpose-reset-window-layouts' ring variable.  • If there is no recent layout, do nothing.
Switch to buffer with purpose	<f11> w P S B</f11>	(purpose-switch-buffer-with- purpose &optional PURPOSE)	Prompt the user and switch to a buffer with purpose PURPOSE.  If called interactively, or with PURPOSE nil, PURPOSE defaults to the current buffer's purpose.
Switch to buffer without taking purpose into account	<f11> w P S b</f11>	(switch-buffer-without-purpose)	Same as C-x b when purpose-mode is not active.
Follow Mode		ws mode which applies all scroll com	mands to all visible windows. To support mouse wheel or scroll bar you need to implement
See also:   Secolling	Text in the first window goes to the bottom and then	continues there.	<ul> <li>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.</li> <li>Follow mode is a minor mode that combines windows into one tall virtual window. This is accomplished by two main techniques: <ul> <li>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.)</li> <li>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.</li> <li>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.</li> </ul> </li> </ul>
Toggle follow-mode See also: ∑ Scrolling	• <f11> w f • <f11>   f</f11></f11>	(follow-mode &optional ARG)	Toggle Follow mode. With a prefix argument ARG, enable Follow mode if ARG is positive, and disable it otherwise.
recentering in current window		o not move point, but reposition the they can be used to refresh the view in	
Position current line to window's Center / Bottom / Top. Refresh screen.	• C-1 • <f11> C-1 * <f7> 5</f7></f11>	(recenter-top-bottom &optional ARG)	Without argument: moves the current line to window: center -> top -> bottom.  • With arg: centre first:  • C-u C-1 C-1 C-1 C-1  • → center → bottom → center → top  • With negative arg: bottom first:  • C C-1 C-1 C-1  • → bottom → center → top  • With arg 0: top first:  • M-0 C-1 C-1 C-1  • → 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-1 key binding because some modes use C-1 as a prefix key.</f11>
Reposition comment/definition in full view	• C-M-1 • C-[ C-1 • Esc C-1	( <b>reposition-window</b> & optional ARG)	Attempts to make the current comment or current definition fully visible by scrolling the lines without changing the point.  Further invocations move it to the top of the window or toggle the visibility of comments that precede it (by scrolling the lines).

## Windows - Reference

Topic/URL	Comment
GNU Emacs — Displaying a Buffer in a Window	Describes the Emacs features related to displaying buffers inside windows.
GNU Emacs Lisp — Displaying Buffers — The Zen of Buffer Display	Describes the rules Emacs tries to use to control the creation of new windows when they are created dynamically from commands.