↖ Topic Index

# Narrowing

| Operation | Keystroke | Function | Note |
|---|---|---|---|
| **Narrow a buffer region**<br><br>See also: ∑ **Buffers** | Emacs narrowing is to restrict display and editing to a portion of a buffer. The remainder of the buffer become temporary inaccessible. If you save the buffer to a file while a portion of the text is narrowed and the rest is invisible, all the buffer content is stored inside the file.<br><br>⚠ This feature can be confusing to new users, so by default the narrow commands are disabled.<br>However, once understood, narrowing becomes very useful. For example, it become possible to perform a non interactive text replacement on a complete portion of the buffer without impacting the rest. It's also easy to concentrate on a single area of the text.<br><br>For all those commands, use **`C-x n w`** to *"widen"* the region back.<br><br>**Notes**:<br>• undo does not widen a narrowed region.<br>• **`(Narrow)`** shows up in the mode line while narrowing is in effect.<br>• Narrowing affects all windows showing the same buffer. If you want to narrow the text inside one window while continuing to see the entire text of the buffer inside another window, you must create an **indirect Buffer** in a separate window. Narrowing inside a buffer does not affect its indirect buffers. See ∑ **Buffers** for more information<br>• Several major modes, like org-mode and markup-mode have extra commands to narrow an area of text. Some minor modes, like lisps-mode, also have commands for narrowing control. Some of these commands are described here but not all. Refer to the documentation of each relevant major mode. | | |
| **Narrow current function definition** | **`C-x n d`** | (**narrow-to-defun** &optional INCLUDE-COMMENTS) | Make text outside current function definition invisible.<br>• The current function definition is the one that contains point or follows point.<br>• Preceding comments are included if INCLUDE-COMMENTS is non-nil.<br>• Works in multiple programming languages (including Elisp, C, Python). |
| **Narrow to marked region** | **`C-x n n`** | (**narrow-to-region** START END) | Restrict editing in this buffer to the current region (the area between mark and point). |
| **Narrow to current page** | **`C-x n p`** | (**narrow-to-page** &optional ARG) | Narrow to current page.<br>If numeric argument supplied, narrows the page identified by the relative number away from current page.<br>☞ Pages in Emacs are identified by ASCII formfeed character (treated as whitespace).<br>☞ To insert a formfeed type: **`C-q C-l`** |
| **Widens the narrowed region** | To return visibility to buffer area that was hidden by the narrow operation, use the following command. | | |
| **Widen a narrowed region** | **`C-x n w`** | (**widen**) | Remove restrictions (narrowing) from current buffer.<br>This allows the buffer's full text to be seen and edited. |
| **Org-mode text structure narrowing** | The following narrowing control commands are used in Org-mode editing.<br>See 𝔐 **Outline/Org-Mode** | | |
| **Narrow current Org block** | **`C-x n b`** | (**org-narrow-to-block**) | Narrow buffer to the current block. Use in Org-mode. |
| **Narrow to current Org element** | **`C-x n e`** | (**org-narrow-to-element**) | Narrow buffer to current Org-mode element. |
| **Narrow to Org sub-tree** | **`C-x n s`** | (**org-narrow-to-subtree**) | Narrow buffer to the current Org-mode subtree. |
| **Markdown mode narrowing** | The following narrowing control commands are used in markdow-mode editing.<br>See 𝔐 **Markdown** | | |
| **Narrow current block** | **`C-x n b`** | (**markdown-narrow-to-block**) | Make text outside current block invisible.<br>• The current block is the one that contains point or follows point |
| **Narrow current subtree** | **`C-x n s`** | (**markdown-narrow-to-subtree**) | Narrow buffer to the current subtree. |
| **Lispy mode narrowing**<br>See 𝔓𝔢- **Lispy** | The following narrowing control commands are used in lispy-mode editing. Used to edit Lisp-family code.<br>📦 This requires the **lispy** external package. PEL downloads, installs and activates lispy when the **pel-use-lispy** user option is set to **t**. | | |
| **Narrow current sexp \| region** | **`N`** | (**special-lispy-narrow** ARG) | Narrow current sexp or region. |
| **Widen** | **`W`** | (**special-lispy-widen**) | Widen back to see the complete buffer. |
| **Narrowing with smartparens** | 📦 The **smartparens** external package is activated by PEL downloads via the **pel-use-smartparens** user-option.<br>• When the smartparens-mode, a minor mode, is activated the following command is used to focus on the current block.<br>• Use **`<f11> i (`** to activate the smartparens-mode. | | |
| **Narrow to sexp**<br><br>See: ∑𝔵 **Smartparens** | **`<M-f7> M-n`** | (**sp-narrow-to-sexp** ARG) | Make text outside current balanced expression invisible.<br>• A numeric arg specifies to move up by that many enclosing expressions.<br>• See also 'narrow-to-region' and 'narrow-to-defun'. |

## Narrowing — Reference

| Topic & link | Description |
|---|---|
| | |
| **GNU Emacs Lisp Manual - Narrowing** | |
| **GNU Emacs Lisp Manual - Point** | |
| | |
| **Emacs: narrow-to-defun, eval-defun, bug @ ErgoEmacs**<br><br>**Old bug - Fixed.** | Describe a bug in narrow-to-defun when the elisp code is not formatted properly, which is part of Xah Lee's complaint that Emacs Lisp (and other Lisp) code should be handled truly semantically and it is not; it must conform to the style guide for several tools to work.<br>He reports the bug in Emacs 24.4.1, 23.x.<br>I tried it in Emacs 26.3 and **Emacs 26.3 narrow-to-defun works fine**. So I assume this issue was fixed since then. |