

# OrgMode Cursor Key Bindings

|             |       | Up                     | Down                  | Left  | Right  |
|-------------|-------|------------------------|-----------------------|---|--|
| shift (⇧)   | S     | org-shiftup            | org-shiftdown         | org-shiftright  | org-shiftright   |
| control (^) | C     | org-backward-paragraph | org-forward-paragraph | (pel-backward-token-start &optional N)  | (pel-forward-token-start &optional N)  |
| ^⇧          | C-S   | org-shiftcontrolup     | org-shiftcontroldown  | org-shiftcontrolleft  | org-shiftcontrolright  |
| meta (⌘)    | M     | org-metaup             | org-metadown          | 1. <i>org-metaleft :: begin of word, stop@-</i><br>2. <i>in terminal: depends on key used:</i><br>1. M- backward-word<br>2. Esc- org-metaleft | 1. <i>org-metaright :: end of word, stop@-</i><br>2. <i>in terminal: depends on key used:</i><br>1. M- forward-word<br>2. Esc- org-metaright |
| ⌘⇧          | M-S   | org-shiftmetaup        | org-shiftmetadown     | org-shiftmetaleft   | org-shiftmetaright   |
| ^⌘          | C-M   | backward-up-list       | down-list             | backward-sexp   | forward-sexp   |
| ^⌘⇧         | C-M-S | backward-up-list       | down-list             | org-decrease-number-at-point  | org-increase-number-at-point   |

## 🚧 Outline Mode & Org-Mode 🚧

| Description  | Keystroke<br>(bold:= global map)   | Function  | Note  |
|--|--|---|---|
| Outline and Org Mode   | 🚧 Emacs Org-Mode is very powerful and contains lots of features. This table only contains some of the basic commands and is far from being complete. |   |   |
| Open this PDF file.<br>See also: <a href="#">🔗 Help/Info</a> | <b>&lt;f11&gt; SPC M-o &lt;f1&gt;</b><br><b>&lt;f12&gt; &lt;f1&gt;</b>   | (pel-help-pdf &optional OPEN-WEB-PAGE)                | Open the <b>M Outline/Org-Mode</b> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around.  |
| <a href="#">🔗 Customize</a> PEL Org Mode support             | <b>&lt;f11&gt; SPC M-o &lt;f2&gt;</b><br><b>&lt;f12&gt; &lt;f2&gt;</b>   | (pel-customize-pel &optional OTHER-WINDOW)            | Customize PEL Org Mode support: open pel-pkg-for-org-mode group. <ul style="list-style-type: none"><li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li></ul>  |
| <a href="#">🔗 Customize</a> Emacs Org Mode support           | <b>&lt;f11&gt; SPC M-o &lt;f3&gt;</b>  | (pel-customize-library &optional OTHER-WINDOW)        | Customize Org Mode external packages support: <ul style="list-style-type: none"><li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li></ul>   |
| Enter Outline mode   | M-x outline-mode   | (outline-mode)  | Set major mode for editing outlines with selective display. <ul style="list-style-type: none"><li>Headings are lines which start with asterisks: one for major headings, two for subheadings, etc. Lines not starting with asterisks are body lines.</li><li>Body text or subheadings under a heading can be made temporarily invisible, or visible again. Invisible lines are attached to the end of the heading, so they move with it, if the line is killed and yanked back. A heading with text hidden under it is marked with an ellipsis (...).</li></ul> |
| List all keys in outline mode                                | C-h f outline-mode <RET>   |   | The org-mode is derived from the outline-mode, so some of the keystrokes from outline-mode are re-used in org-mode  |
| List all keys in org-mode                                    | C-h f org-mode <RET>   |   | Lists the org-mode key mappings. More keys than for outline-mode.   |
| Toggle “clean” view mode.                                    | M-x org-indent-mode  | (org-indent-mode &optional ARG)                       | Toggle use of a <b>cleaner outline view</b> , where: <ul style="list-style-type: none"><li>hides headings leading stars (but still shows in Terminal frames (depending on colors)),</li><li>automatically indent text to align with their headline</li><li>activate proper automatic line wrap.</li></ul> You can activate this minor mode for individual files with:<br><b>#+STARTUP: indent</b>   |
| Control Visibility   |  |   |   |
| Sub-Tree Visibility Cycling                                  | <tab>  | (org-cycle &optional ARG)                             | Expand or compress the set of of <b>sub-tree</b> title lines from the various states: <ul style="list-style-type: none"><li>folded</li><li>see children</li><li>see all subtree</li></ul> To see all: <b>C-u C-u C-u &lt;tab&gt;</b>  |
| Global VisibilityCycling                                     | <b>S-&lt;tab&gt;</b><br><b>C-u &lt;tab&gt;</b>   | (org-shifttab &optional ARG)                          | Rotate the entire buffer among the following states:<br>1) Overview, 2) Contents 3) Show all  |
| Cycle subtree even if it is archived                         | <b>C-&lt;tab&gt;</b>   | (org-force-cycle-archived)                            | <b>Not available in Terminal (termcap) frames.</b>  |
| Reveal context around point                                  | C-c C-r  | (org-reveal &optional SIBLINGS)                       | Show current entry, hierarchy above it, and the following headline.   |
| Show all sub-headings, hide their bodies                     | C-c C-k  | (org-kill-note-or-show-branches)                      | Abort storing current note, or call 'outline-show-branches' to show all subheadings of this heading, but not their bodies.  |
| Show all <i>direct</i> sub-headings                          | C-c <tab>  | (outline-show-children &optional LEVEL)               | If lines (or bodies) are already shown, nothing changes. Does not show the body (but does not hide them when already showing). <ul style="list-style-type: none"><li>By default show only <b>1 level of</b> sub-headings, with numeric argument identify the number of levels to show:<br/>For example to show 3 sub-levels: <b>C-u 3 C-c &lt;tab&gt;</b></li></ul>   |
| Show current subtree in indirect buffer                      | C-c C-x b  | (org-tree-to-indirect-buffer &optional ARG)           | Take current subtree and open in an indirect buffer, not affecting current buffer's visibility. <ul style="list-style-type: none"><li>Any edit in the indirect buffer goes inside the same file.</li><li>Very useful to see the layout and be able to expand information inside a subtree.</li></ul>  |
| Insert headings  |  |   |   |
| Create heading at same level                                 | <b>Esc RET</b><br><b>M-RET</b>   | (org-meta-return &optional ARG)                       | Insert a new heading of same level as current. <ul style="list-style-type: none"><li>If the cursor is at a plain list item, a new item is created.</li><li>If used in the middle of a line, split the line and the rest become a new headline.</li></ul>  |
| TODO Notes   |  |   |   |
| Rotate the TODO State of current item                        | C-c C-t  | (org-todo)  | Changes state in order: unmarked, TODO, DONE. The list of states available is controlled by the <b>org-todo-keywords</b> variable.  |
| Insert a new heading at same level and TODO state            | <b>M-S-RET</b><br><b>C-c C-x M</b>   | (org-insert-todo-heading ARG &optional FORCE-HEADING) | Insert a new heading with the same level and TODO state as current heading. <ul style="list-style-type: none"><li>If the heading has no TODO state, or if the state is DONE, use the first state (TODO by default). Also with one prefix arg, force first state. With two prefix args, force inserting at the end of the parent subtree.</li></ul>  |
| Edit Structure   |  |   |   |
| On a new empty heading: cycle through heading levels         | <tab>  | (org-cycle &optional ARG)                             | This works only if: <ul style="list-style-type: none"><li>the line is empty (except for the star identifying it as a heading)</li><li>there are other headings of level 2 or more somewhere else in the document.</li></ul>   |

| Description   | Keystroke<br>(bold:= global map)   | Function  | Note   |
|---|--|---|--|
| Promote current heading   | <ul style="list-style-type: none"> <li>• <b>M-&lt;right&gt;</b></li> <li>• <b>Esc &lt;right&gt;</b></li> </ul> | (org-metaright &optional ARG)   | <ul style="list-style-type: none"> <li>• In graphics mode: use M-&lt;right&gt;</li> <li>• In Terminal mode: use Esc &lt;right&gt;</li> </ul>   |
| Demote current heading  | <ul style="list-style-type: none"> <li>• <b>M-&lt;left&gt;</b></li> <li>• <b>Esc &lt;left&gt;</b></li> </ul>   | (org-metaleft &optional ARG)  | <ul style="list-style-type: none"> <li>• In graphics mode: use M-&lt;left&gt;</li> <li>• In Terminal mode: use Esc &lt;left&gt;</li> </ul>   |
| Promote current sub-tree  | <b>M-S-&lt;right&gt;</b>   | (org-shiftmetaright)  | Promotes current heading and all headings of its sub-levels.   |
| Demote current sub-tree   | <b>M-S-&lt;left&gt;</b>  | (org-shiftmetaleft)   | Demotes current heading and all headings of its sub-levels.  |
| Move subtree/table row up   | <b>M-&lt;up&gt;</b>  | (org-metaup &optional ARG)  | Everything (the heading and text of the level and all sub-levels) inside the sub-tree moves  |
| Move subtree/table row down   | <b>M-&lt;down&gt;</b>  | (org-metadown &optional ARG)  | Everything (the heading and text of the level and all sub-levels) inside the sub-tree moves  |
| Drag the line at point up   | <b>M-S-&lt;up&gt;</b>  | (org-shiftmetaup &optional ARG)   |  |
| Drag the line at point down   | <b>M-S-&lt;down&gt;</b>  | (org-shiftmetadown &optional ARG)   |  |
| Move through headings   |  |   |  |
| Next heading  | <b>C-c C-n</b>   | (org-next-visible-heading ARG)  | Move to the next visible heading.  |
| Previous heading  | <b>C-c C-p</b>   | (org-previous-visible-heading ARG)  | Move to the previous visible heading.  |
| Next heading same level   | <b>C-c C-f</b>   | (org-forward-heading-same-level ARG &optional INVISIBLE-OK)                 | Move forward to the ARG'th subheading at same level as this one. Stop at the first and last subheadings of a superior heading. Normally this only looks at visible headings, but when INVISIBLE-OK is non-nil it will also look at invisible ones.   |
| Previous heading same level   | <b>C-c C-b</b>   | (org-backward-heading-same-level ARG &optional INVISIBLE-OK)                | Move backward to the ARG'th subheading at same level as this one. Stop at the first and last subheadings of a superior heading.  |
| Backward to higher level heading  | <b>C-c C-u</b>   | (outline-up-heading ARG &optional INVISIBLE-OK)                             | Move to the visible heading line of which the present line is a subheading. With argument, move up ARG levels. If INVISIBLE-OK is non-nil, also consider invisible lines.  |
| Structure Editing   |  |   |  |
| Cut current subtree or region in table  | <b>C-c C-x C-w</b>   | (org-cut-special)   | Cut region in table or cut current subtree.  |
| Paste past subtree or rectangular region into table   | <b>C-c C-x C-y</b>   | (org-paste-special ARG)   | Paste rectangular region into table, or past subtree relative to level.  |
| Hyper-links   |  |   |  |
| Define a link target  | <b>C-c 1</b>   | (org-store-link ARG)  | Store an org-link to the current location. Use this keystroke inside any file at the beginning of a line (including source code)   |
| Insert a hyperlink  | <b>C-c C-1</b>   | (org-insert-link &optional COMPLETE-FILE LINK-LOCATION DEFAULT-DESCRIPTION) | <ul style="list-style-type: none"> <li>• Insert a link. At the prompt, enter the link.</li> <li>• Inside the Org-mode file, insert the link that was created with <b>C-c 1</b>. Use the up arrow to retrieve the link, then type the documentation inside the mini buffer and hit return to insert inside the org mode file.</li> <li>• With a '<b>C-u</b>' prefix, prompts for a file to link to.</li> </ul>  |
| Open the link item  | <b>C-c C-o</b>   | (org-open-at-point &optional ARG REFERENCE-BUFFER)                          | <p>Inside the Org-mode file, move the point over a rendered hyperlink. Type <b>C-c C-o</b> to open a buffer for the target at the location of the target.</p> <p>Some links will open using external software. For example, if the link is a directory, the OS program will be used. To force opening the link inside Emacs use the '<b>C-u</b>' prefix.</p> <p>For example, to open a link to a directory with Dired, use: <b>C-u C-c C-o</b></p> <p>Ref: <a href="https://emacs.stackexchange.com/questions/10426/org-mode-link-to-open-directory-in-dired">https://emacs.stackexchange.com/questions/10426/org-mode-link-to-open-directory-in-dired</a></p>       |
| <p><u>Todo Checkboxes</u></p> <ul style="list-style-type: none"> <li>• TODO items can have checkboxes, written like this: [ ], with a space between the brackets.</li> <li>• There can be a top-level count:             <ul style="list-style-type: none"> <li>• written like this: [ / ], it will show the number of items done over the total (like [24/46], or</li> <li>• written like this: [%], which shows the percentage of items done.</li> </ul> </li> </ul> <p>The following commands are used to control the creation and state of those boxes.</p> |  |   |  |
| <u>Toggle checkbox status</u>   | <b>C-c C-c</b>   | (org-ctrl-c-ctrl-c &optional ARG)   | <p>Toggle checkbox status.</p> <ul style="list-style-type: none"> <li>• With single prefix argument (ie. <b>C-u</b>): toggle checkbox presence at point: add an empty checkbox or remove the current one.</li> <li>• With a double prefix argument (ie. <b>C-u C-u</b>), set it to '[-]', which is considered to be an intermediate state.</li> </ul>  |
| <u>Toggle checkbox status</u>   | <b>C-c C-x C-b</b>   | (org-toggle-checkbox &optional TOGGLE-PRESENCE)                             | <p>Toggle the checkbox in the current line.</p> <ul style="list-style-type: none"> <li>• With prefix argument TOGGLE-PRESENCE, add or remove checkboxes. With a double prefix argument, set the checkbox to "[-]".</li> <li>• When there is an <b>active region</b>, toggle status or presence of the first checkbox there, and make every item inside have the same status or presence, respectively. (This does not seem to work with C-c C-c)</li> <li>• If point is on a headline, apply this to all checkbox items in the text below the heading, taking as reference the first item in subtree, ignoring planning line and any drawer following it.</li> </ul> |
| <u>Insert a new, empty, checkbox on the next line</u>   | <b>M-S-RET</b>   | (org-insert-todo-heading ARG &optional FORCE-HEADING)                       | <p>Insert a new heading with the same level and TODO state as current heading (inserts the checkbox on the current line).</p> <ul style="list-style-type: none"> <li>• If the heading has no TODO state, or if the state is DONE, use the first state (TODO by default). Also with one prefix arg, force first state. With two prefix args, force inserting at the end of the parent subtree.</li> <li>• When called at a plain list item, insert a new item with an unchecked check box.</li> </ul>   |
| Toggle the 'ORDERED' property   | <b>C-c C-x o</b>   | (org-toggle-ordered-property)   | <p>Toggle the 'ORDERED' property of the entry, to toggle if checkboxes must be checked off in sequence.</p> <ul style="list-style-type: none"> <li>• A property is used for this behavior because this should be local to the current entry, not inherited like a tag. However, if you would like to track the value of this property with a tag for better visibility, customize org-track-ordered-property-with-tag.</li> </ul>  |
| Update the statistics cookie  | <b>C-c #</b>   | (org-update-statistics-cookies ALL)   | <p>Update the statistics cookie, either from TODO or from checkboxes.</p> <ul style="list-style-type: none"> <li>• This should be called with the cursor in a line with a statistics cookie. When called with a C-u prefix, update all statistics cookies in the buffer.</li> </ul>  |
| Archiving   |  |   |  |

| Description  | Keystroke<br>(bold:= global map) | Function                                     | Note  |
|--|----------------------------------|--|---|
| Archive current entry                                | <b>C-c C-x C-a</b>               | (org-archive-subtree-default)                | Move the current subtree inside an .org_archive file. The file has the same path and base name.   |
| Archiving Tree                                       | 🔗 Move tree to archive file      |  |   |
| Move current subtree to the archive                  | <b>C-c C-x C-s</b>               | (org-archive-subtree &optional FIND-DONE)    | <p>Move the current subtree to the archive.</p> <ul style="list-style-type: none"> <li>The archive can be a certain top-level heading in the current file, or in a different file. The tree will be moved to that location, the subtree heading be marked DONE, and the current time will be added.</li> <li><b>C-u C-c C-x C-s</b> : When called with a single prefix argument FIND-DONE, find whole trees without any open TODO items and archive them (after getting confirmation from the user).</li> <li><b>C-u C-u C-c C-x C-s</b> : When called with a double prefix argument, find whole trees with timestamps before today and archive them (after getting confirmation from the user). If the cursor is not at a headline when these commands are called, try all level 1 trees. If the cursor is on a headline, only try the direct children of this heading.</li> </ul> |
| Archiving - internal                                 | 🔗 Internal archiving             |  |   |
| Toggle archive tag for current headline              | <b>C-c C-x a</b>                 | (org-toggle-archive-tag &optional FIND-DONE) | <p>Toggle the archive tag for the current headline.</p> <ul style="list-style-type: none"> <li><b>C-u C-c C-x a</b> : With prefix ARG, check all children of current headline and offer tagging the children that do not contain any open TODO items.</li> </ul>  |
| Move the current entry to the <i>Archive Sibling</i> | <b>C-c C-x A</b>                 | (org-archive-to-archive-sibling)             | <p>Archive the current heading by moving it under the archive sibling.</p> <ul style="list-style-type: none"> <li>The archive sibling is a sibling of the heading with the heading name 'org-archive-sibling-heading' and an 'org-archive-tag' tag. If this sibling does not exist, it will be created at the end of the subtree.</li> <li>Archiving time is retained in the ARCHIVE_TIME node property.</li> </ul>   |

## Org-Mode - Reference

| Topic & link  | Description  |
|---|--|
| 🔗 <a href="#">Org mode home</a> and 🔗 <a href="#">The Org Manual</a>  | Org mode is actively developed. The Org Mode home page contains links to all you need. |
| <a href="#">Quick Reference for emacs Org-Mode key binds</a>  |  |
| <a href="#">How to move a subtree to another subtree in org-mode emacs</a>                                  | Stack Overflow Q&A. Just to move the sub-tree itself (not its top level headline).     |
| <a href="#">Command to copy and delete subtree in org-mode</a>  | Stack Overflow Q&A. Answer: C-C C-x C-w and C-c C-x C-y                                |
| <a href="https://www.youtube.com/watch?v=SzA2YODtgK4">https://www.youtube.com/watch?v=SzA2YODtgK4</a>       |  |
| <a href="#">Literate DevOps with Emacs - youtube video</a>  |  |
| <a href="#">David 'Otool Org Tutorial</a>   |  |
| <a href="#">Hello Worg, the Org-Mode Community!</a>   |  |
| <a href="#">Org Mode Tools</a>  |  |
| <a href="#">Org-Mode Exporters - The New Org-mode Exporter Framework</a>                                    |  |
| <a href="#">* ox.el @ github</a>  |  |
| <a href="#">Pandoc support</a>  |  |
| <a href="#">Pandoc-mode - An Emacs mode for interacting with Pandoc</a>                                     |  |
| <a href="#">Org mode and Pandoc</a>   |  |
| <a href="#">Keeping a lab notebook with org-mode, git, Papers, and Pandoc (Part II)</a>                     |  |
| <a href="#">Confluence Wiki Support for Org Mode</a>  |  |
| <a href="https://www.emacswiki.org/emacs/ConfluenceMode">https://www.emacswiki.org/emacs/ConfluenceMode</a> |  |
| <a href="#">confluence-el (10 years old)</a>  |  |
| <a href="#">confluence-el : seems newer.</a>  |  |
| <a href="#">ox-confluence - 2012-2014 : export org files to confluence using the ox.el</a>                  |  |
| <a href="#">sdelafond /org-confluence</a>   | That was integrated inside org-mode.   |
| <a href="#">In org-mode, how do I export to JIRA wiki markup? @ StackExchange</a>                           |  |