

Org-Mode

Description	Keystroke	Function	Note
Org Mode (Org home) - Help & Customize - Text markup / electric keys - Shift-selection - Other markup (◊more) - Clean View Narrow/widen - Visibility Cycling Motion - Edit Structure (+heading) ↵ ↑ copy paste clone sort - TODO state & checkbox - Plain List Operations - Drawers, Notes Properties - HyperLinks - Date & Time +Timestamps - Clocking Work Time - Deadline & Scheduling - Syntax Check Org Buffer - Capture and refile - Archiving - Agenda , Agenda Views - Restriction Locks - Agenda Buffer Commands - View Org File, Chngg Display - Remote Editing	Emacs Org-Mode is very powerful outline document lightweight markup with a lot of powerful features designed for notes, planning and authoring: Markup: <i>*bold*</i>, <i>/italic/</i>, <u>underline</u>, ~code~, =verbatim=, +strike-through+, [[https://orgmode.org][Org links]], LaTeX Document structure: outlines of Headlines with text, Plain Lists, Blocks to encapsulate source code or logging info, Tags and Drawers that hold properties. - Org-mode has commands for document structure editing, visibility cycling, motion through headlines, construction of Sparse Trees, Tables and Spreadsheets. It also has commands to export to other file formats, and includes a publishing management system. You can refile and archive subtrees. - Org mode can create an Agenda with date and time timestamps, time-ranges and deadlines. Time stamps can be active and inactive. It also has Timers. - When clocking time to a task, Org mode displays the task time and name on the ⌘ Mode Line. It can notify on elapsed timer, scheduled events and problems. - Org can also create complex reports based on time clocking information controlled by a clock table (7). ⚠ Some packages conflict with org-mode, PEL provides work around scrolling and window resizing commands by providing key bindings that work in org-mode. See also: ⌘ Calendar GNU Emacs Calculator Org-Mode Reference Card, Org-Mode Command and Function Index, Org Mode CHANGES and previous releases PEL adds key bindings for several Org mode commands that normally do not have any. and adds some other commands. They are accessible under the <f12> key prefix available in Org buffers. Content Display: The way Emacs org-mode displays the content of the Org file depends on its visibility state. It can display everything or hide specific areas of the content. Org mode supports a set of In-Buffer Settings keywords that start with #+ followed by a keyword, a colon, space(s) and words for each setting. To see the heading hierarchy and hide everything else when opening the buffer use: #+STARTUP: content Org Mode v9.8 requires Emacs ≥ 29.1. To upgrade Org on Emacs ≥ 29.1, execute this command from a shell: <pre>emacs -Q -batch -eval "(progn (package-refresh-contents) (package-upgrade 'org))"</pre>		
Last updated on:	2026-09-08	Access org-mode PEL major-mode commands from other buffers with <f11> SPC M-o key prefix (not all shown below).	
Open this PDF file. See also: ⌘ Help/Info	<f11> SPC M-o <f1> <f12> <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the M Org-Mode local PDF. With C-u , M-- , M- -1 or C- -1 prefix: open the remote GitHub hosted PDF instead. If the pel-flip-help-pdf-arg user-option is set it's the other way around.
⌘ Customize PEL Org Mode support	<f12> <f2>	(pel-customize-pel &optional OTHER)	Customize PEL Org Mode support: open pel-pkg-for-org-mode group. If OTHER is non-nil (use C-u), display in another window.
⌘ Customize Emacs Org Mode support	<f12> <f3>	(pel-customize-library &optional OTHER)	Customize Org Mode packages support (and loads them if necessary). If OTHER is non-nil (use C-u), display in another window.
Read Org Info	<f12> ? i	(org-info)	Open the Emacs Info node about org-mode. See ⌘ Help/Info
Shift Select Control See: shift selection See: org-support-shift-select docstring (use <f1> o)	In most major modes, Emacs supports text selection by holding the Shift key down while moving point; this feature is called shift selection . By default, shift selection is not available in org-mode; in org-mode the shift cursor keys are used to control the bullet state, titles and timestamps. 👁👁 However, by you can customize org-support-shift-select to re-activate shift-selection completely (set it to always) or partially (set it to t) which activates shift-selection when the cursor is not in special context, and bullet cycling will only occur when point is exactly over the bullet.		
Text Emphasis Electric Keys QuickStart org-mode emphasis Escape character Org Mode Markup Syntax	👁👁 With the pel-org-use-electric-markup user-option set, PEL provides a specialized electric-pair-mode to type the characters below in pair. 👁👁 The pel-electric-pair-lighter user-option identifies the line-mode lighter displayed when the electric-pair-mode is active. Defaults to “ε(I)” in ⌘ Mode Line For the following keys, and (, {, [: If an area is currently marked, the character typed is inserted before and after the marked area. If no area is marked, two characters (or the matching closing parens) are inserted and the point is left in between.		
Bold	*	To insert a single * character inside the text there are several methods: - Insert a zero-width space right after the asterisk: Type: C-x 8 RET zero width space RET - Insert *~ - Insert == or ~*~ markup	
If you want to use it at the beginning of a line without creating a header, insert a comma right before it; org-mode will strip the comma upon export: insert: , *			
Italic	/	To insert a single / character inside the text there are several methods: - Insert a zero-width space right after the slash: Type: C-x 8 RET zero width space RET - Insert /= or ~/~ markup - Add surrounding spaces: if you put a space before and after the slash it will not trigger italics: example: this / that	
Underline	_	To insert a single _ character inside the text there are several methods: - Insert a zero-width space right after the underscore: Type: C-x 8 RET zero width space RET - Insert =_ = or ~_~ markup - Use a backslash before the underscore if your only concern about is how the document exported to HTML or PDF looks: _	
You may also want to disable subscript for the current file by adding the following configuration line at the top of the Org file. That stops org-mode from turning underscores into subscripts unless you explicitly wrap text in curly braces like this: _[subscript] Add the following entry at the very top of the file and then refresh the buffer by typing C-c C-c C-c on that line after typing it: #+OPTIONS: ^: { }			
Code	~	To insert a single ~ character inside the text there are several methods: - Insert a zero-width space right after the tilde: Type: C-x 8 RET zero width space RET - Insert == markup - Add surrounding spaces: put a space before and after the tilde: example: value ~ 10 - Use a backslash before the tilde if your only concern about is how the document exported to HTML or PDF looks: \~	
Verbatim (literal)	=	To insert a single = character inside the text there are several methods: - Insert a zero-width space right after the equal: Type: C-x 8 RET zero width space RET - Insert ==~ markup - Add surrounding spaces: put a space before and after the equal: example: value = 10	
To use it at the beginning of a line without org-mode interpreting it, insert a comma right before it; org-mode strips the comma upon export: insert: , =			
Strike-trough	+	To insert a single + character inside the text there are several methods: - Insert a zero-width space right after the plus: Type: C-x 8 RET zero width space RET - Insert += or ~+~ markup - Add surrounding spaces: if you put a space before and after the slash it will not trigger italics: example: this + that - Use a backslash before the plus if your only concern about is how the document exported to HTML or PDF looks: \+	
Insert, change or remove emphasis	C-c C-x C-f	(org-emphasize &optional CHAR)	Insert or change an emphasis, specifying the emphasis character or SPACE to remove emphasis. - If there is an active region, change that region to a new emphasis. - If there is no region, just insert the marker characters and position the cursor between them.
Toggle electric-pair-mode in buffer Lighter := ε(I) 	<f11> M-e	(electric-pair-local-mode &optional ARG)	Toggle automatic parens pairing (Electric Pair mode) and org-mode special pair electric keys. With a prefix argument ARG, enable Electric Pair mode if ARG is positive, and disable it otherwise.
Other markup	The following sub-sections describe Org mode extensive markup capabilities. Hyperlink syntax is described in the HyperLinks section.		
Subscripts, Superscripts	‘^’ and ‘_’ are used to indicate super- and subscripts as in: ^superscript _subscript or with surrounding braces: ^{superscript} _{subscript}		
Toggle pretty display works best on GUI Emacs	C-c C-x \	(org-toggle-pretty-entities)	Toggle the composition display of entities as UTF8 characters, in a WYSIWYM way. Works best with GUI based Emacs; terminal mode limits the capabilities and may hide markup.
Paragraphs & line breaks	Paragraphs are separated by empty line(s). Force line break with trailing \\		
Horizontal Rules.	A line consisting of only dashes, and at least 5 of them, is exported as a horizontal line.		
Captions	Assign a caption to a specific part of a document by inserting a ‘CAPTION’ keyword immediately before it: <pre>#+CAPTION: This is the caption for the next table (or link) -----+----- </pre> or use a short caption on one line: <pre>#+CAPTION[Short caption]: Longer caption.</pre>		

Description	Keystroke	Function	Note
Footnotes	Footnotes use the <code>[fn:⊕]</code> where $\oplus$ is either a number or a string. There must be 2 or more of the same set. - Text showing in footnote: follows the <code>[fn:⊕]</code> that starts at the beginning of a line. There must only be one for each $\oplus$. - Other(s) <code>[fn:⊕]</code> are inserted in the text and act as references to the footnote. Each of these footnotes is an org hyperlink. Footnotes are normally inserted inside a Footnotes heading. Control renumbering/sorting after each insertion/deletion with org-footnote-auto-adjust user-option.		
Create footnote, or move from reference to definition (or definition to reference)	C-c C-x f	(org-footnote-action &optional SPECIAL)	- If there is no footnote at point: create one. - If point is at a footnote reference: move to footnote definition. C-c C-o also jumps to it. - If point at a footnote definition: jump to the references if they exist, offer to create them otherwise. - With prefix, or when no footnote can be created, offer additional commands in a menu: Footnotes: [s]ort [r]enumber fn:N [S]=r+s [n]ormalize [d]elete s Sort the footnote definitions by reference sequence. r Renumber the simple ‘fn:N’ footnotes. S Short for first r, then s action. n Rename all footnotes into a ‘fn:1’ ... ‘fn:n’ sequence. d Delete the footnote at point, including definition and references.
Move to footnote definition or reference	C-c C-c	(org-ctrl-c-ctrl-c &optional ARG)	If point is on a footnote reference, jump to definition. If it is at the definition, jump back to the reference. When called at a footnote location with a prefix argument, offer the same menu as C-c C-x f.
	C-c C-o mouse-1/2	(org-open-at-point &optional ARG)	Footnote labels are also links to the corresponding definition or reference, and you can use the usual commands to follow these links.
Special Symbols	org-mode provides <u>LaTeX</u> -like syntax to insert special symbols. Like <code>\alpha</code> for α or <code>\to</code> for $\rightarrow$.		
Show all available special symbols	<f12> ? e	(org-entities-help)	Create a Help buffer with all available entities (special symbols), with their corresponding Org entity markup, LaTeX code and HTML code inside a “Org Entity Help” buffer (which is an org-mode buffer).
Toggle pretty display of special symbol	C-c C-x \	(org-toggle-pretty-entities)	Toggle composition display of entities as UTF8 characters: sub/super script, etc., in a <u>WYSIWYM</u> way. Works in GUI and terminal based Emacs (based on terminal capabilities).
Embedded LaTeX	LaTeX fragments can be present in org mode files: org-mode recognizes LaTeX syntax (also see LaTeX syntax tutorial). LaTeX processing is configured by org-export-with-latex user-option. You can also set it in the file with the following lines: <pre>#+OPTIONS: tex:t Do the right thing automatically (MathJax) #+OPTIONS: tex:nil Do not process LaTeX fragments at all #+OPTIONS: tex:verbatim Verbatim export, for jsMath or so</pre> Previewing LaTeX fragments: use the command below or turn previewing directly in the file with: <pre>#+STARTUP: latexpreview To turn previewing on #+STARTUP: nolatexpreview To turn previewing off</pre>		
Toggle preview of <u>LaTeX</u> fragment at point	C-c C-x C-1	(org-latex-preview &optional ARG)	Toggle preview of the LaTeX fragment at point. - If the cursor is on a LaTeX fragment, create the image and overlay it over the source code, if there is none. Remove it otherwise. - If there is no fragment at point, display images for all fragments in the current section. With an active region, display images for all fragments in the region. - With ‘C-u’ prefix: clear images for all fragments in the current section. - With ‘C-u C-u’ prefix: display image for all fragments in the buffer. - With ‘C-u C-u C-u’ prefix: clear image for all fragments in the buffer.
Insert citation using RefTeX	C-c C-x [	(org-reftex-citation)	Use ‘ reftex-citation ’ to insert a citation into the buffer.
	The line looks like: <code>#+BIBLIOGRAPHY: foo plain option:-d</code> and derives from it that foo.bib is the bibliography file relevant for this document. It then installs the necessary environment for RefTeX to work in this buffer and calls ‘reftex-citation’ to insert a citation into the buffer. Export of such citations to both LaTeX and HTML is handled by the contributed package  ox-bibtex by Taru Karttunen (see Org-mode Contributed Packages).		
Using CDLaTeX 	Enter math with  CDLaTeX minor mode . ( not yet supported by PEL).		
Literal Examples	To write long technical text without escaping any Org-mode markup characters listed above, wrap your text in an Example Block: Org-mode treats everything inside this block as literal, unformatted text. For example: <pre>#+begin_example Here is a long technical note where no markup applies: ,* This is not a header or bold text. A comma MUST precede the * or a #+ at the beginning of a line. / This text will not be italicized. x = y + z_subscript; ~ This is just plain text. #+end_example</pre> For simple examples, you can use lines with a colon followed by spece(s), as in: <pre>Here is an example : Some example from a text file.</pre>		
Source code blocks See also: Org-mode code blocks Structure Templates See (ref:name) syntax and the -r switch to remove the labels from source.	To include source code for a specific programming language use the <code>#+BEGIN_SRC</code> followed by the language major mode name, as in: <pre>#+BEGIN_SRC emacs-lisp (defun org-xor (a b) "Exclusive or." (if a (not b) b)) #+END_SRC</pre> In the example at right: specify line number with -n and use +n to continue line numbers with an optional offset. <pre>#+BEGIN_SRC emacs-lisp -n 20 ;; This exports with line number 20. (message "This is line 21") #+END_SRC #+BEGIN_SRC emacs-lisp +n 10 ;; This is listed as line 31. (message "This is line 32") #+END_SRC</pre>		
Edit source code Terminate/save changes with C-c ' again	C-c '	(org-edit-special &optional ARG) $\rightarrow$ (org-edit-src-code &optional CODE EDIT-BUFFER-NAME)	Edit the <u>HyperLinked</u> source code at point in its native mode in a temporary buffer. The temporary buffer activates the org-src-mode minor mode which binds the C-c ' key to save the changes, close the temporary buffer and go back to the original org-mode buffer.
Insert <u>Structure Templates</u>	C-c C-, <f12> s	(org-insert-structure-template TYPE)	Insert a block structure of the type <code>#+begin_foo/#+end_foo</code> . With an active region, wrap the region in the block. Otherwise, insert an empty block.
	Select a block from ‘ org-structure-template-alist ’ (default shown below) or type either RET, TAB or SPC to write the block type explicitly. a ‘#+BEGIN_EXPORT ascii’ ... ‘#+END_EXPORT’ c ‘#+BEGIN_CENTER’ ... ‘#+END_CENTER’ C ‘#+BEGIN_COMMENT’ ... ‘#+END_COMMENT’ e ‘#+BEGIN_EXAMPLE’ ... ‘#+END_EXAMPLE’ E ‘#+BEGIN_EXPORT’ ... ‘#+END_EXPORT’ h ‘#+BEGIN_EXPORT html’ ... ‘#+END_EXPORT’ l ‘#+BEGIN_EXPORT latex’ ... ‘#+END_EXPORT’ q ‘#+BEGIN_QUOTE’ ... ‘#+END_QUOTE’ s ‘#+BEGIN_SRC’ ... ‘#+END_SRC’ v ‘#+BEGINVERSE’ ... ‘#+ENDVERSE’		
Images and link previews.	Images can be linked. Images are like links: <code>[[./res/image.jpg]]</code> . See HyperLinks section below.  Note that Org-mode 9.8 is available in Emacs 31 and has differs commands for image processing.		
Toggle display of linked images	C-c C-x C-v <f12> v v	(org-toggle-inline-images &optional INCLUDE-LINKED BEG END)	Toggle the display of inline images. INCLUDE-LINKED is passed to ‘org-display-inline-images’.
Refresh inline images	C-c C-x C-M-v <f12> v V	(org-redisplay-inline-images)	Assure display of inline images and refresh them.
Preview image (for Org >= 9.8)	<f12> v r	(org-link-preview-region)	Create inline previews for external links in the active region, or the buffer. With a prefix argument, also preview links with a text description part.
Clear external link previews (for Org >= 9.8)	<f12> v c	(org-link-preview-clear)	Clear external link previews in the active region, or the buffer.

Description	Keystroke	Function	Note
Cleaner Outline View	Org documents are an outline tree. Org headlines (also called headings) start on the left margin with one or more stars followed by space.		
Toggle clean view mode. #+STARTUP: indent 	<f12> <tab>	(org-indent-mode &optional ARG)	Toggle the cleaner outline view, it hides headings leading stars (but still shows in Terminal frames (depending on colors)), automatically indent text to align with their headline and activate proper automatic line wrap. Activate this minor mode for individual files with: #+STARTUP: indent
Narrowing tree/blocks	Limit area of buffer showed in the current window and restrict operations to that area. See ☞ Narrowing for more info.		
Narrow to subtree	C-x n s	(org-narrow-to-subtree &optional ELEMENT)	Narrow buffer to current subtree. • With optional argument ELEMENT narrow to subtree around ELEMENT : for elisp code only.
Narrow to block	C-x n b	(org-narrow-to-block)	Narrow buffer to current block.
Widen	C-x n w	(widen)	Widen buffer to remove narrowing. This allows the buffer's full text to be seen and edited.
Set initial visibility	When opening an org file, all lines are shown by default unless the file has a #+STARTUP: line that identifies the startup folding state: The possible values are: #+STARTUP: overview #+STARTUP: content #+STARTUP: showall		
Visibility Cycling	Org documents are treated as outlines; the heading lines are section levels and can be hidden and shown with the following commands.		
Rotate current subtree between states	<tab>	(org-cycle)	Expand/compress the sub-tree title lines: folded → see children (with drawers) → see all subtree. <i>Works on:</i> a heading line, or: empty line (except for the star identifying it as a heading) there are other headings of level 2 or more somewhere else in the document.
Global Visibility Cycling (only headings, content, all)	• C-u <tab> • S-<tab>	(org-global-cycle)	<i>With point anywhere:</i> Rotate the entire buffer among the following states: 1) Overview, 2) Contents (only headings - <i>useful to see overview</i>) 3) Show all.
Restore property-dependent startup visibility (set with #+STARTUP in file)	C-u C-u <tab>	(org-cycle-set-startup-visibility)	<i>With point anywhere:</i> Restore initial visibility : restores the default property-dependent and startup-configured folding view. Switch back to the startup visibility of the buffer, i.e., whatever is requested by startup options and 'VISIBILITY' properties in individual entries.
Show the whole file, including drawings	C-u C-u C-u <tab>	(org-fold-show-all)	<i>With point anywhere:</i> show all lines, including drawers.
Cycle subtree always	C-c C-<tab>	(org-cycle-force-archived)	Cycle subtree even if it is <i>archived</i> (internally archived). See the Archiving section .
Reveal context around point	C-c C-r	(org-reveal &optional SIBLINGS)	Reveal context around point: show current entry, the following heading, and the hierarchy above. • With 'C-u' prefix: show, on each level, all sibling headings. • With 'C-u C-u' prefix: go to parent, and show the entire tree.
	Useful for working near a location that has been exposed by a Sparse Tree command or an agenda command .		
Show all sub-headings, hide their bodies	C-c C-k	(org-kill-note-or-show-branches)	Expose all the headings of the subtree (show all current subheadings), but not their bodies. Also: abort storing current note.
Show all <i>direct</i> sub-headings	C-c <tab>	(outline-show-children &optional LEVEL)	Expose all direct children of the subtree. If lines (or bodies) are already shown, nothing changes. Does not show the body (but does not hide them when already showing).
	By default show only 1 level of sub-headings, with numeric argument identify the number of levels to show. Example: to show 3 sub-levels: C-u 3 C-c <tab>		
Show current subtree in indirect buffer	C-c C-x b	(org-tree-to-indirect-buffer &optional ARG)	Show the current subtree in an indirect buffer: take current subtree and open in an indirect buffer, not affecting current buffer's visibility. Any edit in the indirect buffer goes inside the same file.
	Very useful to see the layout and be able to expand information inside a subtree. Reuses indirect buffer if it already exists. Close it with: C-x k • With numeric argument N: go up to level N and then take that tree. If N is negative, then go up that many levels. • With 'C-u' prefix: do not remove the previously used indirect buffer.		
Move through headings	See: Motion commands. ARG is 1 by default. Supply a different value via a numeric argument ; with negative value to move in opposite direction.		
Next heading	C-c C-n	(org-next-visible-heading ARG)	Move to the next visible heading. • With numerical argument move that many.
Previous heading	C-c C-p	(org-previous-visible-heading ARG)	Move to the previous visible heading. • With numerical argument move that many.
Next heading same level	C-c C-f	(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.
Previous heading same level	C-c C-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	C-c C-u	(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.
Search/Jump to text	C-c C-j	(org-goto &optional ALTERNATIVE-INTERFACE)	Look-up and jump to a different place without changing the current outline visibility. • With 'C-u' prefix: use alternative ' org-goto-interface ': 'outline' ⇔ 'outline-path-completion'.
	- When you want look-up or go to a different location in a document, the fastest way is often to fold the entire buffer and then dive into the tree. This method has the disadvantage, that the previous location will be folded, which may not be what you want. - This command works around this by showing a copy of the current buffer in an indirect buffer, in overview mode. You can dive into the tree in that copy, use 'org-occur' and incremental search to find a location. When pressing RET or 'Q', the command returns to the original buffer in which the visibility is still unchanged. After RET it will also jump to the location selected in the indirect buffer and expose the headline hierarchy above.		
Edit Structure of document • see: ☞ Scrolling	Modify the outline structure of the Org document by issuing the following commands on heading lines .  With PEL: M-<up> and M-<down> normally bound to PEL <i>scroll commands</i> , but not in org-mode. To scroll up/down in org-mode: use the number keypad key 3 and 9 of the <i>number keypad</i> not num-locked, or when using hydra external package use the <f7> prefix with <up> and <down> keys.		
• Insert headings	Insert outline headings (also called headlines) in the Org document with the following commands. • The org-blank-before-new-entry user-option controls whether a separator empty line is inserted before the new heading.		
Create heading • at same level as line • next heading of parent • end heading of parent	• Esc RET • M-RET	(org-meta-return &optional ARG)	Insert new heading of same level as current . If cursor is at a plain list item, create new item. • If used in the middle of a line, <i>split</i> the line and the rest become a new headline. • With C-u prefix: insert heading after the current sub-tree (next heading of parent tree). • With C-u C-u prefix: insert heading at end of the tree above current heading.
Create heading at same level - don't split line	C-RET	(org-insert-heading-respect-content &optional INVISIBLE-OK)	Insert the line as if cursor is at end of line (never split current line). • This insert heading as if ' org-insert-heading-respect-content ' is currently set to t.
• Insert TODO heading	TODO notes are Org headings that have the TODO state. They can have: priorities , time stamps (scheduling & deadlines) and time tracking .		
Insert a new TODO state heading at same level (split text of current todo) Best to use at end of line.	• M-S-RET • C-c C-x M	(org-insert-todo-heading ARG &optional FORCE-HEADING)	Insert a new heading with the same level and TODO state as current heading. • If the heading has no TODO state, or if the state is DONE, use the first state (TODO by default). • With one prefix arg (C-u): force first state. • With two prefix args (C-u C-u): force inserting at the end of the parent subtree.
Insert a new TODO state heading at same level (never split text)	• C-S-RET • <f12> RET	(org-insert-todo-heading-respect-content &optional ARG)	When point on TODO heading: insert a new TODO heading, never split current line. • This insert heading as if ' org-insert-heading-respect-content ' is currently set to t.
Convert Line <-> Heading	C-c *	(org-ctrl-c-star)	• When point on normal line or plain list item: change line to a subheading at its location. • When point is on a heading: turn into a normal line by removing the stars. If there is an active region, turn all lines in the region into headlines. If the first line in the region was an item, turn only the item lines into headlines. Finally, if the first line is a headline, remove the stars from all headlines in the region.
turn (head)line into item, cycle item type	C-c -	(org-ctrl-c-minus)	With point on plain line: turns a plain line or a region of lines into list items.

Description	Keystroke	Function	Note
Demote current heading (make it a sub-heading)	M-<right>	(org-metaright &optional ARG)	Demote heading, list item at point or move table column right. In front of a drawer or a block keyword, indent it correctly.
Promote current heading	M-<left>	(org-metaleft &optional ARG)	Promote heading, list item at point or move table column left.
Demote current sub-tree	M-S-<right>	(org-shiftmetaright)	Demote subtree or insert table column.
Promote current sub-tree	M-S-<left>	(org-shiftmetaleft)	Promote subtree or delete table column.
Move subtree/table row up (only inside its section)	M-<up>	(org-metaup &optional ARG)	When on heading or plain list item line : move subtree up or move table row up.
Move subtree/table row down (only inside its section)	M-<down>	(org-metadown &optional ARG)	When on heading or plain list item line : move subtree down or move table row down.
Move current line up (only inside its section)	M-S-<up>	(org-shiftmetaup &optional ARG)	When on heading or content line : drag the line at point up, only that line.
Move current line down (only inside its section)	M-S-<down>	(org-shiftmetadown &optional ARG)	When on heading or content line : drag the line at point down only that line.
Cut and paste	See also ↗ Cut & Paste		
Copy subtree	C-c C-x M-w	(org-copy-subtree &optional N CUT FORCE-STORE-MARKERS NOSUBTREES)	Copy the current subtree into the clipboard. A short-hand for marking the subtree and then copying it. - With prefix arg N, copy this many sequential subtrees. - If CUT is non-nil, actually cut the subtree. If FORCE-STORE-MARKERS is non-nil, store the relative locations of some markers in the region, even if CUT is non-nil. This is useful if the caller implements cut-and-paste as copy-then-paste-then-cut.
Copy visible text	C-c C-x v	(org-copy-visible BEG END)	Copy the <i>visible</i> text in the region into the kill ring.
Clone subtree	C-c C-x c	(org-clone-subtree-with-time-shift N &optional SHIFT)	Clone a subtree by making a number of sibling copies of it. Prompts for the number of copies to make; you can also specify whether any timestamps in the entry should be shifted. This can be useful, for example, to create a number of tasks related to a series of lectures to prepare. For more details, see the docstring of the command org-clone-subtree-with-time-shift
Refile and Copy Refile captured note to final destination	C-c C-w	(org-refile &optional ARG DEFAULT-BUFFER RFLOC MSG)	Refile entry or region to a different location. See Refile and Copy . Allows you to move an Org sub-tree or region from one Org file to another Also listed under Capture & Refile section .
Cut subtree	C-c C-x C-w	(org-cut-subtree &optional N)	Cut the current subtree into the clipboard. A short-hand for marking the subtree and then cutting it. With prefix arg N, cut this many sequential subtrees.
Paste subtree	C-c C-x C-y	(org-paste-subtree &optional LEVEL TREE FOR-YANK REMOVE)	Paste the clipboard as a subtree, with modification of headline level. - With 'C-u' prefix: force inserting at the same level as current headline, after subtree at point. - With 'C-u C-u' prefix, force inserting as a child headline, as the first child.
		The entire subtree is promoted or demoted in order to match a new headline level. If the cursor is at the beginning of a headline, the same level as that headline is used to paste the tree before current headline.	
Yank (paste)	C-y	(pel-overwrite-yank &optional ARG)	In org-mode the PEL command invokes (org-yank ARG) to yank like in org-mode: If the kill is a subtree, treat it specially as C-c C-x C-y . See ↗ Cut & Paste for more info.
Mark subtree	C-c @	(org-mark-subtree &optional UP)	Mark the current subtree. Puts point at the start of the current subtree, and mark at the end. With a numeric prefix, move up into the hierarchy of headlines by UP levels before marking the subtree.
Sort entries	C-c ^	(org-sort &optional WITH-CASE)	Sort same-level entries. With an active region, all entries in the region are sorted. Otherwise, the children of the current headline are sorted. With a C-u prefix, sorting is case-sensitive. Prompts for the sorting method, which can be alphabetically, numerically, by time—first timestamp with active preferred, creation time, scheduled time, deadline time—by priority, by TODO keyword—in the sequence the keywords have been defined in the setup—or by the value of a property. Reverse sorting is possible as well. You can also supply your own function to extract the sorting key.
Change TODO items state and priority basic TODO functionality extended use of TODO Progress Logging	Change Org state of headline item with the following commands. The state of TODO items is controlled by the org-todo-keywords user-option. The default is ((sequence "TODO" "DONE")). More complex list are possible, see the variable docstring. When pel-use-org user option is set, PEL sets it to ((sequence "TODO" "IN-PROGRESS" "DONE")) which provides an extra state. 🙌 Several commands are mapped to s-<cursor>. These are only available on headline. In other lines the using the Shift key with cursor is used for shift selection when org-support-shift-select user option is set to t to always.		
Rotate heading state of current item	S-<right>	(org-shiftright &optional ARG)	With point on a headline , cycle switch to the next TODO keyword. empty → TODO → IN PROGRESS → DONE.
	S-<left>	(org-shifleft &optional ARG)	With point on a headline , switch to the next TODO keyword. empty ← TODO ← IN PROGRESS ← DONE.
	C-c C-t		Changes state in order (as in org-todo-keywords) : unmarked, TODO, IN PROGRESS, DONE.
Update state, log progress	C-u C-c C-t	(org-todo)	Prompt for note, record time of the TODO state change. The note is inserted as a list item below the headline, but can also be placed into a drawer. See Progress Logging , Tracking TODO state changes .
Set task priority	C-c ,	(org-priority &optional ACTION SHOW)	Change the priority of an item. Prompts for priority level (A to C). Hit space to remove. With a 'C-u' prefix, show the priority in the minibuffer instead of changing it.
Increase task priority	S-<up>	(org-priority-up)	Increase the priority of the current item.
Decrease task priority	S-<down>	(org-priority-down)	Decrease the priority of the current item.
Todo Checkboxes	- TODO items can have checkboxes, written like this: [] , with a space between the brackets. There can be a top-level count: - written like this: [1] , it will show the number of items done over the total (like [24/46], or - written like this: [%], which shows the percentage of items done.		
Toggle checkbox status • stop clock(s) in DONE item	C-c C-c	(org-toggle-checkbox &optional ARG)	Toggle checkbox status. With ' C-u ' prefix: toggle checkbox presence at point: add/remove checkbox. With 'C-u C-u' prefix, set it to '[-]', which is considered to be an intermediate state.
Toggle checkbox status	C-c C-x C-b	(org-toggle-checkbox &optional TOGGLE-PRESENCE)	Toggle the checkbox in the current line. With prefix argument TOGGLE-PRESENCE, add or remove checkboxes. With a double prefix argument, set the checkbox to "[-]", which is considered to be an intermediate state.
	When there is an active region , toggle status or presence of the first checkbox there, and make every item inside have the same status or presence, respectively. If point on a headline, apply to all checkbox items in the text below heading, taking as reference the 1st item in subtree, ignoring planning line & any drawer after.		
Toggle radio button	C-c C-x C-r	(org-toggle-radio-button &optional ARG)	Toggle checkbox status by using the checkbox of the item at point as a radio button: when the checkbox is turned on, all other checkboxes on the same level will be turned off. - With C-u prefix: toggle the presence of the checkbox. - With C-u C-u prefix: set it to '[-]', which is considered to be an intermediate state.
Insert a new, empty, checkbox on the next line	M-S-RET	(org-insert-todo-heading ARG &optional FORCE-HEADING)	Insert a new heading with the same level and TODO state as current heading (inserts the checkbox on the current line). If heading has no TODO state, or if state is DONE, use the first state (default: TODO). When called at a plain list item , insert a new item with an unchecked check box.
	With C-u prefix: force first state. With C-u C-u prefix: force inserting at the end of the parent subtree.		
Toggle the ' ORDERED ' property	C-c C-x o	(org-toggle-ordered-property)	Toggle the ' ORDERED ' property of the entry, to toggle if checkboxes must be checked off in sequence. 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.
Update the statistics cookie	C-c #	(org-update-statistics-cookies ALL)	With point on line with statistic cookie : update statistics cookie, either from TODO or from checkboxes. When called with a C-u prefix, update all statistics cookies in the buffer.

Description	Keystroke	Function	Note
Plain Lists • Unordered list items → • Ordered list items → • Description list items →	Within an entry of the outline tree, hand-formatted lists can provide additional structure. They also provide a way to create lists of checkboxes. Org supports editing such lists, and every exporter can parse and format them. Org knows ordered lists, unordered lists, and description lists: • start with ‘-’, ‘+’, or ‘*’ as bullets (to use ‘*’ the line must be indented to prevent mistaking it to an headline. • start with a numeral followed by either a period or a right parenthesis, such as ‘1.’ or ‘1)’ If you want a list to start with a different value—e.g., 20—start the text of the item with ‘[@20]’. • Those constructs can be used in any item of the list in order to enforce a particular numbering. • are unordered list items, and contain the separator ‘::’ to distinguish the description term from the description text. The following commands are available to work on plain list items.		
Cycle visibility of plain list items	<tab>	(org-cycle &optional ARG)	Items can be folded just like headline levels. Normally this works only if point is on a plain list item. For more details, see the variable org-cycle-include-plain-lists. • If this variable is set to <i>integrate</i>, plain list items are treated like low-level headlines. • The level of an item is then given by the indentation of the bullet/number. Items are always subordinate to real headlines, however; the hierarchies remain completely separated. • In a new item with no text yet, the first TAB demotes the item to become a child of the previous one. Subsequent TABs move the item to meaningful levels in the list and eventually get it back to its initial position.
Insert new item at current level	M-RET	(org-insert-item &optional CHECKBOX)	Insert a new item at the current level. • If cursor is before first character after bullet of the item, the new item will be created before the current one. • If CHECKBOX is non-nil, add a checkbox next to the bullet.
Insert new item with unchecked box	• M-S-RET • <f12> M-RET	(org-insert-todo-heading ARG &optional FORCE-HEADING)	With point at a plain list item: insert a new item with an unchecked check box.
Jump to previous item (if shift select is off)	S-<up>		With point at a plain list item: Jump to the next item in the current list. • Only if org-support-shift-select is off. Otherwise it shift-selects the text.
Jump to next item (if shift select is off)	S-<down>		With point at a plain list item: Jump to the next item in the current list. • Only if org-support-shift-select is off. Otherwise it shift-selects the text.
Move to beginning of previous paragraph/item	C-<up>	(org-backward-paragraph &optional ARG)	Move backward by a paragraph, or equivalent, unit. • With argument ARG, do it ARG times; a negative argument ARG = -N means move forward N paragraphs. The function moves point between two structural elements (paragraphs, tables, lists, etc.).
Move to beginning of next paragraph/item	C-<down>	(org-forward-paragraph &optional ARG)	Move forward by a paragraph, or equivalent, unit. • With argument ARG, do it ARG times; a negative argument ARG = -N means move backward N paragraphs. The function moves point between two structural elements (paragraphs, tables, lists, etc.) 👉 This command and the previous one also provides the following special moves for convenience: • on a table or a property drawer, move to its beginning; • on comment, example, export, source and verse blocks, stop at blank lines; • skip consecutive clocks, diary S-exps, and keywords.
Move item (and its subtree) up	M-<up>	(org-move-item-up)	With point at a plain list item: Move the item at point up, i.e. swap with previous item. • Sub-items (items with larger indentation) are considered part of the item, so this really moves item trees.
Move item (and its subtree) down	M-<down>	(org-move-item-down)	With point at a plain list item: Move the item at point down, i.e. swap with following item. • Sub-items (items with larger indentation) are considered part of the item, so this really moves item trees.
Outdent item (not its children)	M-<left>	(org-outdent-item)	With point at a plain list item: Outdent a local list item, but not its children. • If a region is active, all items inside will be moved.
Indent item (not its children)	M-<right>	(org-indent-item)	With point at a plain list item: Indent a local list item, but not its children. • If a region is active, all items inside will be moved.
Outdent item and sub-items	M-S-<left>	(org-outdent-item-tree)	With point at a plain list item: Outdent a local list item including its children. • If a region is active, all items inside will be moved.
Indent item and sub-items	M-S-<right>	(org-indent-item-tree)	With point at a plain list item: Indent a local list item including its children. • If a region is active, all items inside will be moved.
Toggle state of item checkbox	C-c C-c	(org-ctrl-c-ctrl-c &optional ARG)	With point at a plain list item checkbox: toggle the state of the checkbox. • Also verify bullets and indentation consistency in the whole list.
Toggle the list bullets	C-c -	(org-cycle-list-bullet &optional WHICH)	With point on the first line of a plain list item: Cycle through the different itemize/enumerate bullets of the entire list. • This cycle the entire list level through the sequence: ‘-’ -> ‘+’ -> ‘*’ -> ‘1.’ -> ‘1)’ • If WHICH is a valid string, use that as the new bullet. If WHICH is an integer, 0 means ‘-’, 1 means ‘+’ etc. If WHICH is ‘previous’, cycle backwards.
Toggle heading ↔ plain list item	C-c *	(org-toggle-heading &optional NSTARS)	• With point on heading: Convert headings to normal text • With point at a plain list item: Convert items or text to headings.
Convert plain list into heading subtree	• C-c C-* • <f12> 1 *	(org-list-make-subtree)	Convert the plain list at point into a subtree. • Turn the whole plain list into a subtree of the current heading. Checkboxes become ‘TODO’ or ‘DONE’ keywords when unchecked or checked, respectively.
Cycle list item bullet	• S-<left> • S-<right>		With point on list item bullet: cycles bullet styles when point is on the bullet. • Or anywhere in an item line, with details depending on org-support-shift-select.
Sort plain list items	C-c ^	(org-sort-list &optional WITH-CASE SORTING-TYPE GETKEY-FUNC COMPARE-FUNC INTERACTIVE?)	On a plain list item: Sort the plain list. Sublists are not sorted. Checkboxes, if any, are ignored. • Prompt for the sorting method: numerically, alphabetically, by time, or by custom function. • Comparing entries ignores case by default. However, with an optional argument WITH-CASE, the sorting considers case as well, if the current locale allows for it.
	Sorting is identified by the following prompt entry: Capital letters reverse the sort order.	n Numerically, by converting the beginning of the item to a number. a Alphabetically. Only the first line of item is checked. t By date/time, either the first active time stamp in the entry, if any, or by the first inactive one. • In a timer list, sort the timers. x By "checked" status of a check list. f specifies a function to be called with point at the beginning of the record. • It must return a value that is compatible with COMPARE-FUNC, the function used to compare entries.	
Repair plain list	<f12> 1 r	(org-list-repair)	Fix indentation, bullets and checkboxes in the list at point.

Description	Keystroke	Function	Note
Drawers and Notes	Org mode has drawers to keep information associated with an entry which is not normally shown. They can contain anything but a headline and another drawer.		
Insert drawer at point Ideally issue this at end of line.	C-c C-x d	(org-insert-drawer &optional ARG DRAWER)	Insert a drawer at point. Prompt for drawer name. Point is left between drawer's boundaries. If a region is active, insert the drawer around that region instead. When optional argument ARG is non-nil, insert a property drawer.
Add note	C-c C-z	(org-add-note)	Add a time-stamped note to the current entry. Done in the same way as adding a state change note.
Properties	Org properties are key-value pairs associated with a heading and inserted inside its drawer. See Property Syntax and the Column View section . To insert a property: Type : at the beginning of a line or type C-c C-x p Properties cannot be inserted to a Plain List item, meant to be lightweight.		
Select property key	M-TAB	(pcomplete-completions-at-point)	After an initial colon in a line , complete property keys. All keys used in the current file are offered as possible completions. - Provide standard completion using pcomplete's completion tables. - Same as 'pcomplete' but using the standard completion UI.
Insert/set a property	:		When typed at the beginning of a line: insert a property.
	C-c C-x p	(org-set-property PROPERTY VALUE)	Prompt for a property name, offering completion on existing and default properties. And then prompt for a value, offering completion either on allowed values (via an inherited xxx_ALL property) or on existing values in other instances of this property in the current file. Insert markup for that property key-value pair.
Add/Set a property	C-c C-c s	(org-set-property PROPERTY VALUE)	When point is over a property: add/set PROPERTY to VALUE in the current entry. Prompt for property name (with completion) and value (with completion) on this property in current file.
Delete property from current entry	C-c C-c d	(org-delete-property PROPERTY)	When point is over a property: delete PROPERTY of the current entry. Prompts for the property.
Delete property globally	C-c C-c D	(org-delete-property-globally PROPERTY)	When point is over a property: remove PROPERTY globally, from all entries. Prompts for the property. This function ignores narrowing, if any.
Compute property at point	C-c C-c c	(org-compute-property-at-point)	When point is over a property: compute the property at point. It looks for an enclosing column format, extracts the operator and then applies it to the property in the column format's scope.
	For example: given a tree of heading items with an <code>:Effort:</code> property that identifies number of hours, issuing the command in the parent level heading would compute the sum of all <code>:Effort:</code> values identified in the <code>:effort:</code> properties of child heading items and store it in the parent heading <code>:Effort:</code> property value.		
Switch to next allowed value	S-<right>	(org-property-next-allowed-value)	When point is over a property key tag: switch to the next allowed value for this property.
Switch to previous allowed value	S-<left>	(org-property-previous-allowed-value)	When point is over a property key tag: switch to the previous allowed value for this property.
Hyperlinks	Org mode supports links in the file to external files, web pages, emails and much more. Link format: <code>[[LINK][DESCRIPTION]]</code> or <code>[[LINK]]</code> It also supports internal links , links to other parts of the current file/buffer such as: <code>[[#my-custom-id]]</code> to entry with CUSTOM_ID property or <code>[[*Some section]]</code> to the headline with the name 'Some Section'. <code><<My Target>></code> to any element that has an element name using the following format: <code>#+NAME: My Target</code>		
Store a link to the current location (define a link target) Video: linking files @ 🧠	C-c l	(org-store-link ARG &optional INTERACTIVE?)	Store a link to the current location in 'org-stored-links' that can later be inserted into an Org buffer with 'org-insert-link' (C-c C-1). 🗨 Type these keys at the beginning of a line to get a proper link.
	<f12> . - A 'C-u' prefix: negates 'org-link-context-for-files' for file links or 'org-gnus-prefer-web-links' for links to Usenet articles. - A 'C-u C-u' prefix: forces skipping storing functions that are not part of Org core. - A 'C-u C-u C-u' prefix: forces storing a link for each line in the active region.		
Insert a hyperlink or Edit the visible LINK part of an already formatted link.	C-c C-1	(org-insert-link &optional COMPLETE-FILE LINK-LOCATION DEFAULT-DESCRIPTION)	Insert a link at point. Supports several types of links; select type with tab-completion, then complete it and hit <RET> . Then enter the description text and hit <RET> . Inside the Org-mode file, insert the link that was created with C-c l or <f12> l . 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. With a 'C-u' prefix, prompts for a file to link to.
Open/follow the link item Example: open directory in Dired	C-c C-o	(org-open-at-point &optional ARG REFERENCE-BUFFER)	With point over a rendered hyperlink: open a buffer for the target at the location of the target. - Some links will open using external software. For example, if the link is a directory, the OS program will be used. - With 'C-u' prefix: force opening the link inside Emacs or other window For example, to open a link to a directory with Dired, use: C-u C-c C-o
Record a position in mark ring	C-c %	(org-mark-ring-push &optional POS BUFFER)	Put the current position into the mark ring and rotate it. Later, after some movement, you can use C-c & to go back to the saved position.
Go back	C-c &	(org-mark-ring-goto &optional N)	Jump to previous position in mark ring. With prefix arg N, jump back that many stored positions. When called several times in succession, walk through the entire ring. Org mode commands jumping to a different position in current file, or to another Org file, automatically push old position onto the ring.
Move to next hyperlink	C-c C-x C-n	(org-next-link &optional SEARCH-BACKWARD)	Move forward to the next link. If the link is in hidden text, expose it. When SEARCH-BACKWARD is non-nil, move backward.
Move to previous hyperlink	C-c C-x C-p	(org-previous-link)	Move backward to the previous link. If the link is in hidden text, expose it.
Dates and Times	To assist project planning, TODO items can be labeled with a date and/or a time: an org timestamp . Format: <code><2003-09-16 Tue 12:00-12:30></code>		
• Creating Timestamps	Time/Date range: 2 timestamps connected by '-'. Format: <code><2004-08-23 Mon 10:00-11:00>--<2004-08-26 Thu 10:00-11:00></code>		
Add/replace active time stamp • To insert a Time/Date range issue the command twice.	C-c .	(org-timestamp ARG &optional INACTIVE)	Insert/replace an active time stamp. Prompts. By default only the date is included. - With 'C-u' prefix: the time stamp contains date & time. - With 'C-u C-u' prefix: the time stamp contains date & time. No prompt.
	All parts of a date not specified by the user are filled in from the timestamp at point, if any, or the current date/time otherwise.		
Add/replace inactive time stamp	C-c !	(org-timestamp-inactive &optional ARG)	Insert an inactive time stamp. Prompts. By default only the date is included. - With 'C-u' prefix: the time stamp contains date & time. - With 'C-u C-u' prefix: the time stamp contains date & time. No prompt. Replace existing timestamp if there was any.
	An inactive time stamp is enclosed in square brackets instead of angle brackets. It is inactive in the sense that it does not trigger agenda entries. So these are more for recording a certain time/date.		
Normalize timestamp	C-c C-c	(org-ctrl-c-ctrl-c &optional ARG)	When point on a timestamp: normalize timestamp, insert/fix day name if missing or wrong
Insert date timestamp from calendar current position	C-c <	(org-date-from-calendar)	Insert time stamp corresponding to cursor date in "Calendar" buffer. If there is already a time stamp at the cursor position, update it.
Change timestamp date	S-<right>	(org-shiftright &optional ARG)	With point on timestamp: increment date. Use numeric arg to increment by that number.
	S-<left>	(org-shiftright &optional ARG)	With point on timestamp: decrement date. Use numeric arg to decrement by that number.
Go to calendar at date specified by timestamp	C-c >	(org-goto-calendar &optional ARG)	Go to the Emacs calendar at the current date. - If there is a time stamp in the current line, go to that date, otherwise go to todays' date. - With 'C-u' prefix: force the current date.
Open agenda at date	C-c C-o	(org-open-at-point &optional ARG)	When point on a timestamp: open the agenda at the day specified.
Evaluate time range	C-c C-y	(org-evaluate-time-range &optional TO-BUFFER)	Evaluate a time range by computing the difference between start and end. - Print result in the echo area, but with prefix arg insert result after time stamp in buffer. - If the time range is in a table, insert result into the next column. ⚠ A year is assumed to be exactly 365 days in order to avoid rounding problems.

Description	Keystroke	Function	Note
• Clocking Work Time Check-in & check-out time is logged into the :LOGBOOK: drawer See next section below to create a time tracking report.	Org mode allows tracking time for each task item; by clocking in and clocking out, adding logged timestamps in the item drawer to track task time. • Resolving idle time: org mode supports a mechanism to handle being away from Emacs for a while; idle time. It provides a command to resolve the situation and either terminate the open periods with some time, none or keep it as if all was fine and update the logged time stamps accordingly. • To activate automatic idle time set the org-clock-idle-time user-option to the number of minutes of idling to get an alert when idle periods exceed that time. • Continuous clocking: to start clocking from the last task clock-out time. To activate systematically, set org-clock-continuously to non nil. When continuous clocking is active, when you clock-in Org retrieves the clock-out time of the last clocked entry fir this session and starts the new clock from there. • Clocking out automatically after some idle time: when you often forget to clock out before being idle, and don't want to adjust clocking times manually, you can set org-clock-auto-clockout-timer to a number of seconds and add (org-clock-auto-clockout-insinuate) to your '.emacs' file. • With PEL, just set the pel-org-clock-auto-clockout-timer customizable user-option to the number of seconds.		
Clock IN: Start clock on current item	C-c C-x C-i	(org-clock-in &optional SELECT START-TIME)	Start the clock on the current item. If necessary, clock-out of the currently active clock. • With 'C-u' prefix: offer a list of recently clocked tasks to clock into. • With 'C-u C-u C-u' prefix: start clocking using the last clock-out time, if any: for Continuous clocking
Clock OUT: Stop clock on current item	C-c C-x C-o	(org-clock-out &optional SWITCH-TO-STATE FAIL-QUIETLY AT-TIME)	Stop the currently running clock. • Throw an error if there is no running clock and FAIL-QUIETLY is nil. • With a universal prefix (C-u), prompt for a state to switch the clocked out task to, overriding the existing value of 'org-clock-out-switch-to-state'.
CANCEL the current clock (remove all its current time)	C-c C-x C-q	(org-clock-cancel)	Cancel the running clock by removing the start timestamp.
Restart clock: start clock on last closed clocked item	C-c C-x C-x	(org-clock-in-last &optional ARG)	Clock in the last closed clocked item. When already clocking in, send a warning. • With 'C-u' prefix: select the task you want to clock in from the last clocked in tasks. • With 'C-u C-u' prefix: start clocking using the last clock-out time, if any: for Continuous clocking • With 'C-u C-u C-u' prefix: prompt for a todo state to switch to, overriding the existing value 'org-clock-in-switch-to-state'.
Recompute time interval	C-c C-y	(org-evaluate-time-range &optional TO-BUFFER)	Evaluate a time range by computing the difference between start and end. • Print in the echo area. With prefix arg TO-BUFFER, insert result in buffer after the date stamp. • If the time range is actually in a table, the result is inserted into the next column. • For time difference computation, a year is assumed to be exactly 365 days in order to avoid rounding problems.
Resolve any half-open clocks See: Resolving idle time	C-c C-x C-z	(org-resolve-clocks &optional ONLY-DANGLING-P PROMPT-FN LAST-VALID)	Resolve all currently open Org clocks. • If 'only-dangling-p' is non-nil, only ask to resolve dangling (i.e., not currently open and valid) clocks.
Set effort property See: Effort Estimates	C-c C-x e	(org-set-effort &optional INCREMENT VALUE)	When point over a heading line or any of its children text/plain list items: Set entrie's effort property. • If INCREMENT is non-nil, set the property to the next allowed value. Otherwise, if optional argument VALUE is provided, use it. Prompt for the new value if none of the previous variables is set.
Modify/update Effort Estimates	C-c C-x C-e	(org-clock-modify-effort-estimate &optional VALUE)	Modify effort estimate of the item currently being clocked: add to or set the effort estimate of the item currently being clocked. VALUE can be a number of minutes, or a string with format hh:mm or mm. • When the string starts with a + or a - sign, the current value of the effort property will be changed by that amount. • If the effort value is expressed as an unit defined in 'org-duration-units' (e.g. "3h"), the modified value will be converted to a hh:mm duration. This command will update the "Effort" property of the currently clocked item, and the value displayed in the mode line.
Increment timestamp element	S-<up>	(org-timestamp-up &optional ARG)	With point over timestamp: increase/decrease the date item at the cursor by one. With point over the year, change the year, the month, the day or the time, change that. With point on the enclosing bracket, change the timestamp type. • With prefix ARG, change by that many units. • Call 'org-timestamp-up' or 'org-timestamp-down' .
Decrement timestamp element	S-<down>	(org-timestamp-down &optional ARG)	
Increment date	S-<right>	(org-shiftright &optional ARG)	
Decrement date	S-<left>	(org-shifleft &optional ARG)	With point over timestamp: increase/decrease the date (only the date). • With prefix ARG, change by that many units.
Move point to specified TODO item	C-c C-x C-j	(org-clock-goto &optional SELECT)	Go to the currently clocked-in entry, or to the most recently clocked one. • With prefix arg SELECT (C-u), offer recently clocked tasks for selection. • Type identified character to move to specified item.
Display clock times for headlines in current file	C-c C-x C-d	(org-clock-display &optional ARG)	Show subtree times in the entire buffer. By default, show the total time for the range defined in 'org-clock-display-default-range'. • With 'C-u' prefix: show the total time for today instead. • With 'C-u C-u' prefix: use a custom range, entered at prompt. • With 'C-u C-u C-u' prefix: display the total time in the echo area.
Generate a report for clock activity	C-c C-x C-r	(org-toggle-radio-button &optional ARG)	Toggle off all checkboxes and toggle on the one at point.
Create sparse tree(s) from TODO items See Sparse Trees • Useful to hide parts of document that you don't need to see and focus on other parts.	C-c /	(org-sparse-tree &optional ARG TYPE)	Create a sparse tree, prompt for the details. This command can create sparse trees. You first need to select the type of match used to create the tree: t Show all TODO entries as a sparse tree. T Show entries with a specific TODO keyword. m Show entries selected by a tags/property match. p Enter a property name and its value (both with completion on existing names/values) and show entries with that property. r Show entries matching a <i>regular expression</i> ('/' can be used as well). b Show deadlines and scheduled items before a date. a Show deadlines and scheduled items after a date. d Show deadlines due within 'org-deadline-warning-days'. D Show deadlines and scheduled items between a date range.
Go to next sparse tree	• M-g n • M-g M-n	(next-error)	
Go to previous sparse tree	• M-g p • M-g M-p	(previous-error)	Jump to the previous sparse tree match in this buffer.
• The clock table	Produce reports based on time clocking information using a <i>clock table</i> definition syntax supporting custom-made elisp code. See clock table in References		
Create/Update clock report • Also type C-c C-c on the +=BEGIN line	<f12> r	(org-clock-report &optional ARG)	Update or create a table containing a report about clocked time. • If point is inside an existing clocktable block, update it. Otherwise, insert a new one. • With 'C-u' prefix: move to the first clock table in the buffer and update it. • The new table inherits its properties from the variable 'org-clock-clocktable-default-properties'. The scope of the clocktable, when not specified in the previous variable, is 'subtree' of the current heading when the function is called from inside heading, and 'file' elsewhere (before the first heading).
Shift current :block: interval & update the table	• S-<right> • S-<left>	(org-clocktable-try-shift DIR N)	Shift the current ':block' interval and update the table. Point needs to be in the '#+=BEGIN: clocktable' line for this command. If ':block' is 'today', it is shifted to 'today-1', etc.
Use Dynamic Blocks to create/maintain a Time Tracking Report	Once time log (as defined in Clocking Work Time above) is setup in the Org file you can create and maintain tabular time tracking report with the following commands. 💡 See PEL's example/templates/org-mode/master-org.org file if you want to create a system to track your projects tasks. It has the required clock table setup and has instructions on how to use. It's best to open this file with Emacs: it has links to important user-options you will need to set.		
Create clocking REPORT Dynamic Block	C-c C-x x	(org-dynamic-block-insert-dblock TYPE &optional INTERACTIVE-P)	Insert a dynamic block of type TYPE. • Prompts for TYPE: use 'clocktable' to create an work time report. Supports completion & history. • Selects the dynamic block types among defined types, per 'org-dynamic-block-define'.
Update dynamic block (update clock report) See also: <f12> t M-=	C-c C-x C-u	(org-dblock-update &optional ARG)	With point over dynamic block: User command for updating dynamic blocks. • With 'C-u' prefix: update all dynamic blocks in the buffer.
	C-c C-c		With point over dynamic block: Update the dynamic block at point.

Description	Keystroke	Function	Note											
Deadlines and Scheduling	A timestamp may be preceded by special keywords to facilitate planning; the <code>timestamp</code> and keyword must be positioned immediately after the task they refer to.													
	Deadline:	*** TODO write article about the need for vision, competence and empathy in all world leaders DEADLINE: <2026-11-2 Mon> The editor in charge is [[bbdb:Louise Arbour]]												
	Scheduled task:	*** TODO Call Bill Gates for AI industry regulation strategy. SCHEDULED: <2026-08-27 Thu>												
		- Once a scheduled task is marked done it will still be listed on that date, unless you set the org-agenda-skip-scheduled-if-done user-option. - A scheduled item represents a "do-date" to start working on a task, which rolls over daily in the agenda until completed, whereas a plain timestamp denotes a fixed appointment or event. A scheduled task acts as a reminder for action, while a timestamped event merely indicates a specific point in time. For instance, a project report would be marked with SCHEDULED: <date> to begin work, while a team meeting would use a plain timestamp (<date>) to mark a singular event.												
	Repeated tasks	Some tasks need to be repeated again and again. Org mode helps to organize such tasks using a so-called repeater in a ‘DEADLINE’ , ‘SCHEDULED’ , or plain timestamps												
Insert a deadline	C-c C-d	(org-deadline ARG &optional TIME)	Insert a "DEADLINE:" string with a timestamp to make a deadline. Pops calendar to select date. - With ‘C-u’ prefix: Remove any deadline from the item. - With ‘C-u C-u’ prefix: Prompt for a warning delay.											
	With argument TIME, set the deadline at the corresponding date. TIME can either be an Org date like "2011-07-24" or a delta like "+2d".													
Insert a scheduled time	C-c C-s	(org-schedule ARG &optional TIME)	Insert a "SCHEDULED:" string with a timestamp to schedule an item. Pops calendar to select date. - With ‘C-u’ prefix: Remove any scheduling date from the item. - With ‘C-u C-u’ prefix: Prompt for a delay cookie.											
	With argument TIME, scheduled at the corresponding date. TIME can either be an Org date like "2011-07-24" or a delta like "+2d".													
Show past-due deadlines	C-c / d	(org-check-deadlines)	Create a sparse tree with all deadlines that are either past-due, or which will become due within org-deadline-warning-days . With ‘C-u’ prefix: show all deadlines in the file.											
	With a numeric prefix: check that many days. To show all deadlines due tomorrow type: C-1 C-c / d or, with PEL In terminal mode type: M-1 C-c / d													
Show past-due deadlines and scheduled items	C-c / b	(org-check-before-date)	Show deadlines and scheduled items before a date. Pops up the Emacs calendar to let the user select a date.											
Show deadlines and scheduled items after a date	C-c / a	(org-check-after-date)	Show deadlines and scheduled items after a date. Pops up the Emacs calendar to let the user select a date.											
Org Syntax Check	Tools to check the syntax of org files.													
Lint org file report warnings	<f12> c	(org-lint)	Check current Org buffer for syntax mistakes. By default, run all checkers.											
	With a ‘C-u’ prefix ARG, select one category of checkers only. With a ‘C-u C-u’ prefix, run one precise checker by its name. All lint warnings are shown in the "Org Lint" buffer. In that buffer the following commands are available: <table> <tr> <td>C-j, TAB</td><td>Display the offending line</td><td>h</td><td>Hide all reports from the same checker</td></tr> <tr> <td>RET</td><td>Move point to the offending line</td><td>i</td><td>Also remove them from all subsequent checks</td></tr> <tr> <td>g</td><td>Check the document again</td><td>s</td><td>Sort reports by the column at point</td></tr> </table>			C-j , TAB	Display the offending line	h	Hide all reports from the same checker	RET	Move point to the offending line	i	Also remove them from all subsequent checks	g	Check the document again	s
C-j , TAB	Display the offending line	h	Hide all reports from the same checker											
RET	Move point to the offending line	i	Also remove them from all subsequent checks											
g	Check the document again	s	Sort reports by the column at point											
Parse org file	(org-element-parse-buffer &optional GRANULARITY VISIBLE-ONLY KEEP-DEFERRED)		Recursively parse the buffer and return structure. If narrowing is in effect, only parse the visible part of the buffer. See output in the "Message" buffer.											
Switch buffer to an Org buffer See: Buffers	<f11> b o <f11> M-o b	(org-switchb &optional ARG)	Switch between Org buffers. <f11> M-o b is only available once an org file is opened. - With ‘C-u’ prefix: restrict available buffers to files. - With ‘C-u C-u’ prefix: restrict available buffers to agenda files.											
Capture Using capture	Capture lets you quickly store notes with little interruption of your work flow. Captured text is stored in the file identified by org-default-notes-file user-option; it defaults to ~/.notes. - Once text is captured in the note file, you can refile it inside the locations identified by org-refile-targets user-option (use <f11> M-o <f4> r to access it).													
Capture something into the Org notes file.	<f11> M-o c	(org-capture &optional GOTO KEYS)	Select a template from ‘org-capture-templates’, and then file the newly captured information. The text is immediately inserted at the target location, and an indirect buffer is shown where you can edit it. Pressing ‘C-c C-c’ brings you back to the previous state of Emacs, to continue your work.											
	- With a ‘C-u’ prefix argument GOTO: don’t capture anything, just go to the file/headline where the selected template stores its notes. - With a ‘C-u C-u’ prefix argument: go to the last note stored. When called with a ‘C-0’ (zero) prefix, insert a template at point. - When called with a ‘C-1’ (one) prefix, force prompting for a date when a datetree entry is made. - Instead of Control key for numerical prefixes you can also use the Meta key. If ‘org-capture-use-agenda-date’ is non-nil, capturing from the agenda will use the date at point as the default date. Then, a ‘C-1’ prefix will tell the capture process to use the HH:MM time of the day at point (if any) or the current HH:MM time.													
Finalize capture	C-c C-c	(org-capture-finalize)	When finished entering information into the capture buffer: C-c C-c returns you to the window configuration before the capture process, so that you can resume your work without further distraction. With a prefix argument, finalize and then jump to the captured item											
Refile and Copy Refile captured note to final destination	C-c C-w	(org-capture-refile)	Finalize the current capture and then refile the entry. Move subtree or region from one file to another. - Refiling is done from the base buffer, because the indirect buffer is then already gone. - Any prefix argument will be passed to the refile command.											
Kill capture	C-c C-k	(org-capture-kill)	Abort the current capture process.											
Customize user-option that identifies where to refile notes	<f11> M-o <f4> r	(pel-org-set-refile-targets &optional OTHER-WINDOW)	Customize ‘org-refile-targets’ . 🐸 <f11> M-o <f4> r is only available once an org file is opened. With a prefix argument, open customize buffer in other window.											
Archiving	Archiving is the process of moving finished tasks or old subtrees out of your active working file. An archive is where those hidden items go so they do not clutter your daily view or slow down your agenda searches, while still remaining saved for future reference.													
Archive current entry	C-c C-x C-a	(org-archive-subtree-default)	Move the current subtree inside an .org_archive file. The file has the same path and base name.											
Move tree to archive file	The most common archiving action is to move a project tree to another file, called the archive file.													
Move current subtree to the archive	C-c C-x C-s C-c \$	(org-archive-subtree &optional FIND-DONE)	Move the current subtree to the archive. 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.											
Archive only trees with: - done items → - items expiring in past →	- With ‘C-u’ prefix: find whole trees without any open TODO items and archive them (after getting confirmation from the user). - With ‘C-u C-u’ prefix: 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.													
Restore archived tree back to its original Org file	<f12> M-a r	(pel-org-archive-restore &optional SILENT)	Restore archived sub-tree at point or before point back to its original org file. Signal an error when failing to restore item back to its original Org file unless SILENT is non-nil.											
Internal archiving	If you want to just switch off—for agenda views—certain subtrees without moving them to a different file, you can use the ‘ARCHIVE’ tag. This not moving text to an archive file; it is simply hidden.													
Toggle archive tag for current headline	C-c C-x a	(org-toggle-archive-tag &optional FIND-DONE)	Toggle the archive tag for the current headline. With ‘C-u’ prefix: check all children of current headline and offer tagging the children that do not contain any open TODO items.											
Move the current entry to the <i>Archive Sibling</i>	C-c C-x A	(org-archive-to-archive-sibling)	Archive the current heading by moving it under the archive sibling. - 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. - Archiving time is retained in the ARCHIVE_TIME node property.											

Description	Keystroke	Function	Note
Org Agenda	With all scheduled tasks identified inside one or several Org files, org mode can display agendas for that work.		
• Agenda Files	Org mode tracks the files that contain task information that must be taken into account in your agenda. These files are listed in the org-agenda-files user-option.		
Move/Add current file to top agenda file list	C-c [	(org-agenda-file-to-front &optional TO-END)	Move/add the current file to the top/front of the agenda file list. With optional argument TO-END, add/move to the end of the list
Remove Org file from agenda file list	C-c]	(org-remove-file &optional FILE)	Remove current file from the list of files in variable ‘org-agenda-files’.
Cycle agenda files	<f12> a , - C-’ - C-,	(org-cycle-agenda-files)	Cycle through the files in ‘org-agenda-files’. - If the current buffer visits an agenda file, find the next one in the list. - If the current buffer does not, find the first agenda file.
Agenda Views	You may have several TODO items, time-stamped items, and tagged headlines over one or several files, as defined by the agenda files (described just above). Org can collect all this information and present it inside a unified agenda view that you can easily manage.		
- Org Agenda - Agenda Views - Agenda Commands org-agenda prompts for the command key shown to the right.	<f12> a a C-c a <f11> M-o a	(org-agenda &optional ARG KEYS RESTRICTION)	Open the agenda view: dispatch agenda commands to collect entries to the agenda buffer. - Prompts for a command to execute. - Any prefix arg will be passed on to the selected command. n Agenda and all TODOs a Agenda for current week or day / Multi-occur t List of all TODO entries m Match a TAGS/PROP/TODO query s Search for keywords ? Find :FLAGGED: entries * Toggle sticky agenda views < Buffer, subtree/region restriction > Remove restriction e Export agenda views T Entries with special TODO kwd M Like m, but only TODO entries S Like s, but only TODO entries C Configure custom agenda commands # List stuck projects (!=configure) If the current buffer is in Org mode and visiting a file, you can also first press ‘<’ once to indicate that the agenda should be temporarily, until the next use of <f12> restricted to the current file. Pressing ‘<’ twice means to restrict to the current subtree or region (if active)
• Restriction locks	Org mode restriction locks limit your agenda scope to a specific file or subtree. - They prevent the global agenda from pulling in tasks from unrelated files, helping you focus strictly on a local context. - Issue the following commands from org-mode files or agenda view buffers. - You can also use the < and > keys in the Speedbar to restrict the agenda on an item or remove the restriction.		
Set restriction lock	C-c C-x <	(org-agenda-set-restriction-lock &optional TYPE)	Set restriction lock for agenda to current subtree or file. When in a restricted subtree, remove it. The restriction will span over the entire file if TYPE is ‘file’, or if TYPE is (4), or if the cursor is before the first headline in the file. Otherwise, only apply the restriction to the current subtree.
Remove restriction lock	C-c C-x >	(org-agenda-remove-restriction-lock &optional NOUPDATE)	Remove agenda restriction lock.
Opening agenda view	The following command, issued from within an org-mode buffer allows opening the agenda view for specific date		
Open agenda for date at point	C-c C-o	(org-open-at-point &optional ARG)	With point on a timestamp: open agenda for the date at cursor
Agenda buffer commands	The following commands are available inside the agenda view buffer only; in org-agenda-mode buffers.		
⌘ Customize org-agenda	<f12> <f3>	(pel-customize-org-agenda &optional OTHER)	Customize Org Agenda (customization group) (and loads library if necessary). If OTHER is non-nil (use C-u), display in another window.
Next line	- n - C-n <down>	(org-agenda-next-line)	Move cursor to the next line, and show if follow mode is active.
Previous line	- p - C-p <up>	(org-agenda-previous-line)	Move cursor to the previous line, and show if follow mode is active.
• View Org File			
Display original item in Org file (in another buffer)	SPC	(org-agenda-show-and-scroll-up &optional ARG)	Display the Org file which contains the item at point. When called repeatedly, scroll the window that is displaying the buffer. With ‘C-u’ prefix: display the item, but fold drawers.
Display org file for agenda item at point in other buffer	L	(org-agenda-recenter ARG)	Display the Org file which contains the item at point in agenda view and recenter.
Goto original item location in other window	TAB	(org-agenda-goto &optional HIGHLIGHT)	Go to the entry at point in the corresponding Org file.
Goto original item location in org file; display in current buffer.	RET	(org-agenda-switch-to &optional DELETE-OTHER-WINDOWS)	Go to the Org mode file which contains the item at point. When optional argument DELETE-OTHER-WINDOWS is non-nil, the displayed Org file fills the frame.
Toggle follow mode	F	(org-agenda-follow-mode)	Toggle follow mode in an agenda buffer. In Follow mode, as you move point through the agenda buffer, the other window always shows the corresponding location in the Org file.
Show current item subtree inside another buffer.	C-c C-x b	(org-agenda-tree-to-indirect-buffer ARG)	Show the subtree corresponding to the current entry in an indirect buffer. - This calls the command ‘org-tree-to-indirect-buffer’ from the original buffer. - With numeric argument N: go up to level N and then take that tree. If N is negative, then go up that many levels. - With ‘C-u’ prefix: make a separate frame for this tree, i.e. don’t use the dedicated frame.
• Change display	The “Org Agenda” buffer shows the ISO 8601 week number .		
Select another agenda view; append it to current view	A	(org-agenda-append-agenda)	Append another agenda view to the current one: allows interactive building of block agendas. Agenda views are separated by ‘org-agenda-block-separator’.
Close all other windows	o	(delete-other-windows &optional WINDOW INTERACTIVE)	Close all other windows.
	C-x 1		
Update agenda view	v	(org-agenda-view-mode-dispatch)	Prompt for the command (shown below) and invoke the selected command (described in the next rows).
	View: [d]ay [w]eek for[t]night [m]onth [y]ear [SPC]reset [q]uit/abort time: [G]rid [I]nactive [f]ollow [l]og [L]og-all [c]lockcheck [a]rch-trees [A]rch-files clock[R]eport include[D]iary [E]ntryText		
Switch to default view	v SPC	(org-agenda-reset-view)	Restore default view for agenda (a week). The org-agenda-span user-option identifies the default.
Switch to day view	d	(org-agenda-day-view &optional DAY-OF-MONTH)	Switch to daily view for agenda. With argument DAY-OF-MONTH, switch to that day of the month.
	v d		
Switch to week view	w	(org-agenda-week-view &optional ISO-WEEK)	Switch to weekly view for agenda. With argument ISO-WEEK, switch to the corresponding ISO week.
	v w		
Switch to fortnight view (2 weeks)	v t	(org-agenda-fortnight-view &optional ISO-WEEK)	Switch to fortnightly view for agenda: 14 days. With argument ISO-WEEK, switch to the corresponding ISO week.
		If ISO-WEEK has more then 2 digits, only the last two encode the week. Any digits before this encode a year. So 200712 means week 12 of year 2007. Years ranging from 70 years ago to 30 years in the future can also be written as 2-digit years.	

Description	Keystroke	Function	Note
Switch to month view	v m	(org-agenda-month-view &optional MONTH)	Switch to monthly view for agenda. - With argument MONTH, switch to that month. If MONTH has more then 2 digits, only the last two encode the month. Any digits before this encode a year. So 200712 means December year 2007. - Years ranging from 70 years ago to 30 years in the future can also be written as 2-digit years.
Switch to year view	v y	(org-agenda-year-view &optional YEAR)	Switch to yearly view for agenda. - With argument YEAR, switch to that year. - Years ranging from 70 years ago to 30 years in the future can also be written as 2-digit years.
Toggle log mode	l	(org-agenda-log-mode &optional SPECIAL)	Toggle log mode in an agenda buffer. - With argument SPECIAL, show all possible log items, not only the ones configured in ‘org-agenda-log-mode-items’. - With ‘C-u C-u’ prefix: show *only* log items, nothing else.
	• v l • v L		
Toggle Diary	D	(org-agenda-toggle-diary)	Toggle diary inclusion in an agenda buffer.
	v D		
Toggle time Grid	G	(org-agenda-toggle-time-grid)	Toggle time grid in an agenda buffer.
	v G		
Toggle display of habits	K	(org-habit-toggle-display-in-agenda ARG)	Toggle display of habits in agenda. See ‘ org-habit-show-all-today ’ user-option. With ARG toggle display of all vs. undone scheduled habits.
Toggle text entry mode	E	(org-agenda-entry-text-mode &optional ARG)	Toggle entry text mode in an agenda buffer. This mode displays a configurable number of lines from the body of the original Org heading directly beneath each agenda line, providing quick inline context without forcing you to jump to the source file.
	v E		
Toggle clock table display mode	R	(org-agenda-clockreport-mode)	Toggle clocktable mode in an agenda buffer.
	v R		
Toggle inclusion of archive items	v a	(org-agenda-archives-mode &optional WITH-FILES)	Toggle inclusion of items in trees marked with :ARCHIVE:. When called with a prefix argument, include all archive files as well
Toggle inclusion of archive items and files	v A		Toggle inclusion of items in trees marked with :ARCHIVE: including all archive files.
Rebuild agenda view	r	(org-agenda-redo &optional ALL)	Rebuild possibly ALL agenda view(s) in the current buffer.
Rebuild agenda all view(s)	g	(org-agenda-redo-all &optional EXHAUSTIVE)	Rebuild all agenda views in the current buffer. With a prefix argument, do so in all agenda buffers.
Filter entries with regard to their tags	/	(org-agenda-filter &optional STRIP-OR-ACCUMULATE)	Prompt for a general filter string and apply it to the agenda. - With ‘C-u’ prefix: negate the entire filter, useful in connection with the prompt history. - With ‘C-u C-u’ prefix: add the new filter elements to the existing ones (instead of + in string) - With ‘C-u C-u C-u’ prefix: filter according to ‘org-agenda-auto-exclude-function’.
			- The string may contain filter elements like: +category +tag +<effort +>effort +=effort +/regexp/ - Instead of ‘+’, ‘-’ is allowed to strip the agenda of matching entries. ‘+’ is optional if it is not required to separate two string parts. Multiple filter elements can be concatenated without spaces, for example: +work-Pierre<0:10-/plot/ selects entries with category ‘work’ and effort estimates below 10 minutes, and deselects entries with tag ‘Pierre’ or matching the regexp ‘plot’. - During entry of the filter, completion for tags, categories and effort values is offered. Since the syntax for categories and tags is identical there should be no overlap between categories and tags. If there is, tags get priority.
Archive entry/sub-tree	a	(org-agenda-archive-default-with-confirmation)	Archive the entry or subtree belonging to the current agenda entry
Show next period	f	(org-agenda-later ARG)	Go forward in time by the current span in the agenda buffer. With prefix ARG, go forward that many times the current span.
Show previous period	b	(org-agenda-earlier ARG)	Go backward in time by the current span in the agenda buffer. With prefix ARG, go backward that many times the current span.
Show today's info	.	(org-agenda-goto-today)	Go to today’s date in the agenda buffer.
Show specified date info	j	(org-agenda-goto-date DATE)	Show specified date. When called interactively, prompt for the date. When called from Lisp, DATE should be a date as returned by ‘org-read-date’.
• Remote editing	Emacs Org-mode remote editing commands let you change, update, and manage your original .org task files directly from the global agenda buffer without opening those source files. This keeps your master files synchronized and prevents data drift		
Undo remote Org file change See ⌘ Undo/Redo/Repeat	• C-_- • C-M-_ • M-u • <f11> u u	(org-agenda-undo)	Undo a remote editing step in the agenda. This undoes changes both in the agenda buffer and in the remote buffer that have been changed along. ⚠ org-agenda-mode does not support undoing and undo (to essentially redo). To do that you must go back to the original Org file (use <tab> for that) and then undo/redo the change from there. ⚠ PEL does not support C-_ binding; PEL uses it as a prefix key.
Cycle todo state	t	(org-agenda-todo &optional ARG)	On TODO line: Cycle TODO state of line at point, also in Org file. This changes the line at point, all other lines in the agenda referring to the same tree node, and the headline of the tree node in the Org file.
Switch agenda entry to previous TODO	• C-S-<left> • <f12> <up>	(org-agenda-todo-previousset)	Switch TODO entry to previous sequence.
Switch agenda entry to next TODO	• C-S-<right> • <f12> <down>	(org-agenda-todo-nextset)	Switch TODO entry to next sequence.
Kill Org entry/subtree	C-k	(org-agenda-kill)	Kill the entry or subtree belonging to the current agenda entry
Show item tags	T	(org-agenda-show-tags)	Show the tags applicable to the current item.
Set headline tag	• : • C-c C-q	(org-agenda-set-tags &optional TAG ONOFF)	Set tags for the current headline.
Set item priority	• , • C-c ,	(org-agenda-priority &optional FORCE-DIRECTION)	Set the priority of line at point, also in Org file. This changes the line at point, all other lines in the agenda referring to the same tree node, and the headline of the tree node in the Org file. - Called with one universal prefix arg, show the priority instead of setting it. - When called programmatically, FORCE-DIRECTION can be ‘set’, ‘up’, ‘down’, or a character.
Increase item priority	• + • S-<up> • C-c C-x <up>	(org-agenda-priority-up)	Increase the priority of line at point, also in Org file.
Decrease item priority	- S-<down> C-c C-x <down>	(org-agenda-priority-down)	Decrease the priority of line at point, also in Org file.
Set effort for item	• C-c C-x e • e	(org-agenda-set-effort)	Set the effort property for the current headline.
Add note to entry	• C-c C-z • z	(org-agenda-add-note &optional ARG)	Add a time-stamped note to the entry at point.
Schedule the item	C-c C-s	(org-agenda-schedule ARG &optional TIME)	Schedule the item at point. ARG is passed through to ‘org-schedule’.

Description	Keystroke	Function	Note
Jump to running clock in other window	J	(org-agenda-clock-goto)	Jump to the currently clocked in task within the agenda. If the currently clocked in task is not listed in the agenda buffer, display it in another window.
Change timestamp to next day	- S-<right> - C-c C-x <right>	(org-agenda-do-date-later ARG) - With a numeric prefix: change it by that many days. For example, 3 6 5 S-RIGHT changes it by a year. - With ‘C-u’ prefix: change the time by one hour. If you immediately repeat the command, it will continue to change hours even without the prefix argument.. - With ‘C-u C-u’ prefix: do the same for changing minutes. The stamp is changed in the original Org file, but the change is not directly reflected in the agenda buffer. Use x or g to update the buffer.	Change the timestamp associated with the current line by one day into the future. If the date is in the past, the first call to this command moves it to today.
Change timestamp to past day	- S-<left> - C-c C-x <left>	(org-agenda-do-date-earlier ARG)	Change the timestamp associated with the current line by one day into the past.
Change timestamp of item	>	(org-agenda-date-prompt ARG)	Change the date of this item. Date is prompted for, with default today. The prefix ARG is passed to the ‘org-timestamp’ command and can therefore be used to request time specification in the time stamp
Clock-in on item	- I - C-c C-x C-i	(org-agenda-clock-in &optional ARG)	Start the clock on the currently selected item.
Clock-out	- O - C-c C-x C-o	(org-agenda-clock-out)	Stop the currently running clock.
Cancel running clock	- X - C-c C-x C-x	(org-agenda-clock-cancel &optional ARG)	Cancel the currently running clock.
Attachment Commands	C-c C-a	(org-attach)	The dispatcher for attachment commands. Shows a list of commands and prompts for another key to execute a command.
	a Select a file and attach it to the task, using ‘org-attach-method’. c Attach a file using copy method. m Attach a file using move method. l Attach a file using link method. y Attach a file using symbolic-link method. u Attach a file from URL (downloading it). b Select a buffer and attach its contents to the task. n Create a new attachment, as an Emacs buffer. z Synchronize the current node with its attachment directory, in case you added attachments yourself. o Open current node's attachments. O Like "o", but force opening in Emacs. f Open current node's attachment directory. Create if missing. F Like "f", but force using Dired in Emacs. d Delete one attachment, you will be prompted for a file name. D Delete all of a node's attachments. A safer way is to open the directory in dired and delete from there. s Set a specific attachment directory for this entry. Sets DIR property. S Unset the attachment directory for this entry. Removes DIR property. q Abort.		
Capture, use date at point	k	(org-agenda-capture &optional WITH-TIME)	Call ‘org-capture’ with the date at point. With a ‘C-1’ or M-1 prefix, use the HH:MM value at point (if any) or the current HH:MM time.
Bulk remote editing selected entries	Instead of changing tasks one by one, "bulk" editing lets you process a group of tasks simultaneously. - You mark multiple tasks across the agenda view (using commands like org-agenda-bulk-mark). - You execute one single command that applies to all of them at once.		
Mark an entry for bulk action	m	(org-agenda-bulk-mark &optional ARG)	Mark entries for future bulk action. - When ARG is nil or one and region is not active then mark the entry at point. - When ARG is nil or one and region is active then mark the entries in the region. - When ARG is greater than one mark ARG lines.
Mark all entries for bulk action	*	(org-agenda-bulk-mark-all)	Mark all entries for future agenda bulk action.
Unmark entry at point	u	(org-agenda-bulk-unmark &optional ARG)	Unmark the entry at point for future bulk action.
Unmark all marked	U	(org-agenda-bulk-unmark-all)	Remove all marks in the agenda buffer. This removes the markers and the overlays.
Toggle mark at point	M-m	(org-agenda-bulk-toggle)	Toggle the mark at point for bulk action.
Toggle all marks	M-*	(org-agenda-bulk-toggle-all)	Toggle all marks for bulk action.
Toggle mark of entries matching regexp	%	(org-agenda-bulk-mark-regexp REGEXP)	Mark entries matching REGEXP for future agenda bulk action.
Bulk action Accepts the following:	B p Toggle persistent marks. \$ Archive all selected entries. A Archive entries by moving them to their respective archive siblings. t Change TODO state. This prompts for a single TODO keyword and changes the state of all selected entries, bypassing blocking and suppressing logging notes—but not timestamps. + Add a tag to all selected entries. - Remove a tag from all selected entries. s Schedule all items to a new date. To shift existing schedule dates by a fixed number of days, use something starting with double plus at the prompt, for example ‘++8d’ or ‘++2w’. d Set deadline to a specific date. r Prompt for a single refile target and move all entries. The entries are no longer in the agenda; refresh (g) to bring them back. S Reschedule randomly into the coming N days. N is prompted for. With a prefix argument (C-u B S), scatter only across weekdays. f Apply a function to marked entries. For example, the function below sets the ‘CATEGORY’ property of the entries to ‘web’.	(org-agenda-bulk-action &optional ARG)	Execute an remote-editing action on all marked entries. Prompts for action: The prefix arg is passed through to the command if possible.
Using Column view in the Agenda	Column view (see Column View) is normally used to view and edit properties embedded in the hierarchical structure of an Org file. It can be quite useful to use column view also from the agenda, where entries are collected by certain criteria.		
Column view in Agenda	C-c C-x C-c	(org-agenda-columns)	Turn on or update column view in the agenda.
Exporting Agenda Views	Export agenda to plain text, HTML, Postscript, PFD, iCalendar files		
Export agenda to another file format	C-x C-w	(org-agenda-write FILE &optional OPEN NOSETTINGS AGENDA-BUFNAME)	Write the current buffer (an agenda view) as a file. Selects file format from file extension: plain text (.txt),HTML (.html or .htm), PDF (.pdf) or Postscript (.ps) is produced. If the extension is .ics, translate visible agenda into iCalendar format. If the extension is .org, collect all subtrees corresponding to the agenda entries and add them in an .org file.
Calendar commands	See also ⓘ Calendar		
Open calendar of date at point	c	(org-agenda-goto-calendar)	When in Org agenda: Open the Emacs calendar with the date at the cursor.
In calendar: show Org Agenda	c	(org-calendar-goto-agenda)	When in the calendar: Compute and show the Org agenda for the date at point.

Description	Keystroke	Function	Note
Agenda diary	An Emacs Org mode agenda diary is an integrated feature that pulls traditional Emacs diary entries—such as holidays, anniversaries, and recurring appointments—directly into your Org mode weekly or daily agenda view. Unified View: Combines your active task deadlines and scheduled items with personal calendar events from a standard Emacs diary file. Special Expressions: Supports S-expression commands like <code>%%(org-anniversary . . .)</code> to track birthdays & age milestones dynamically inside your files. Holiday Integration: Displays regional or custom holidays seamlessly alongside work projects  To enable this: org-agenda-include-diary user-option must be turned on		
Insert new entry into the agenda diary	i	(org-agenda-diary-entry)	Insert a new entry into the diary, using the date at point and (for block entries) the date at the mark.
	This adds to the Emacs diary file, in a way similar to the i command in the calendar. The diary file pops up in another window, where you can add the entry. - If you configure org-agenda-diary-file to point to an Org file, Org creates entries in that file instead. Most entries are stored in a date-based outline tree that will later make it easy to archive appointments from previous months/years. The tree is built under an entry with a 'DATE_TREE' property, or else with years as top-level entries. - Emacs prompts you for the entry text—if you specify it, the entry is created in org-agenda-diary-file without further interaction. If you directly press RET at the prompt without typing text, the target file is shown in another window for you to finish the entry there. See also the <code>k r</code> command		
Show the phases of the moon	M	(org-agenda-phases-of-moon)	Show the phases of the moon for the three months around current date.
Show sunrise and sunset	S	(org-agenda-sunrise-sunset)	Show sunrise and sunset times. The geographical location must be set with calendar variables, see the documentation for the Emacs calendar .
Show holidays	H	(org-agenda-holidays)	Show holidays for three months around point date.
Convert date	C	(org-agenda-convert-date)	Convert the date at point into many other cultural and historic calendars.
Quit and exit			
Quit agenda	q	(org-agenda-quit)	Quit agenda, remove the agenda buffer. - When 'org-agenda-sticky' is non-nil, bury the agenda buffer instead of killing it. - When 'org-agenda-restore-windows-after-quit' is non-nil, restore the pre-agenda window configuration. - When column view is active, exit column view instead of the agenda.
Exit agenda	x	(org-agenda-exit)	Exit agenda, remove the agenda buffer and all buffers loaded by Emacs for the compilation of the agenda. Buffers created by the user to visit Org files are not removed. Like 'org-agenda-Quit', but kill any buffers that were created by the agenda. Org buffers visited directly by the user will not be touched. Also, exit the agenda even if it is in column view.
Column View	In column view, each heading is turned into a table row. Columns in the table provide access to the properties of the entries. Also work in Agenda View mode. See: Defining columns: Scope of columns definitions , Column attributes		
Using Column View	Turning column view on or off		
Activate column view	C-c C-x C-c	(org-columns &optional GLOBAL COLUMNS-FMT-STRING)	Turn on column view on an Org mode file. Column view applies to the whole buffer if point is before the first headline. Otherwise, it applies to the first ancestor setting "COLUMNS" property. If there is none, it defaults to the current headline. - With a 'C-u' prefix argument, GLOBAL, turn on column view for the whole buffer unconditionally. - When COLUMNS-FMT-STRING is non-nil, use it as the column format.
Recreate column view	r g	(org-columns-redo)	When point on a column view line : Construct the column display again.
Exit column view	q C-c C-c	(org-columns-quit)	When point on a column view line : Remove the column overlays and exit column editing.
Editing Values	To navigate up down and left right inside the columns use the cursor keys.		
Select nth allowed value	1..9, 0		Directly select the Nth allowed value, 0 selects the 10th value.
	For this to do anything, you must have predefined allowed values set up for that property (e.g., <code>#+PROPERTY: Effort_ALL 0:10 0:30 1:00 2:00</code>). If your cursor is on an Effort field and you press 3, Org Mode will automatically insert 1:00 into that cell.		
Change value to next allowed value	n S-<right>	(org-columns-next-allowed-value &optional PREVIOUS NTH)	Switch to the next allowed value for this column. When PREVIOUS is set, go to the previous value. When NTH is an integer, select that value.
Change value to previous allowed value	p S-<left>	(org-columns-previous-allowed-value)	Switch to the previous allowed value for this column.
Edit property at point	e	(org-columns-edit-value &optional KEY)	Edit the value of the property at point in column view. Where possible, use the standard interface for changing this line.
Show column value	v	(org-columns-show-value)	Show the full value of the property: show the value in the echo area. This is useful if the width of the column is smaller than that of the value.
Edit list of allows values	a	(org-columns-edit-allowed)	Edit the list of allowed values for the current property. If the list is found in the hierarchy, the modified values are stored there. If no list is found, the new value is stored in the first entry that is part of the current column view.
Toggle checkbox or quit	C-c C-c	(org-columns-toggle-or-columns-quit)	Toggle checkbox at point, or quit column view.
Modify column view	 The column view is an interactive overlay presentation. You cannot undo the deletion of a column after having issued the command to delete the column. The quickest way to restore it is to exit column view and re-enter it (q followed by C-c C-x C-c). This recalculates the layout from the buffer's original properties.		
Decrease columns width	<	(org-columns-narrow ARG)	Make the column narrower by ARG characters.
Increase columns width	>	(org-columns-widen ARG)	Make the column wider by ARG characters.
Insert new column	S-M-<right>	(org-columns-new &optional SPEC &rest ATTRIBUTES)	Insert a new column, to the left of the current column. - Interactively fill attributes for new column. When column format specification SPEC is provided, edit it instead. - When optional argument attributes can be a list of columns specifications attributes to create the new column non-interactively. See 'org-columns-compile-format' for details.
Delete current column	S-M-<left>	(org-columns-delete)	Delete the column at point from columns view.
Move the current column right	M-<right>	(org-columns-move-right)	Swap this column with the one to the right.
Move the current column left	M-<left>	(org-columns-move-left)	Swap this column with the one to the left.
Moving rows	 Contrary to the other commands that modify the column view, moving rows can be undone with M-u (and other bindings to undo) - Note that a row moves up or down at its current outline level.  When crossing hierarchies, a row can get "<i>swallowed</i>", as Org Mode tries to preserve the entire tree structure (the headline, its text, and all of its sub-items) as a single block when moving it. The Boundary Problem: If you move a Level 2 heading (**) downward and it passes the very last Level 2 heading of that section, it will jump past the Level 1 parent (*) below it. Suddenly, your Level 2 heading is now grouped under a completely different Level 1 category. Hidden Subheadings: Column view summarizes data. If you have collapsed subheadings that you cannot see in your current column view, moving a row might accidentally move those invisible child subheadings along with it, shifting your data structure more than intended.		
Move the current row up	M-<up>	(org-columns-move-row-up)	Move the current table row up.
Move the current row down	M-<down>	(org-columns-move-row-down)	Move the current table row down.
Capturing Column View	Since column view is just an overlay over a buffer, it cannot be exported or printed directly. If you want to capture a column view, use a 'columnview' dynamic block (see Dynamic Blocks in page 6).		

Description	Keystroke	Function	Note									
Timers	Org provides two types of timers. - There is a relative timer that counts up, which can be useful when taking notes during, for example, a meeting or a video viewing. - There is also a countdown timer.											
Start or reset the relative timer (counts up)	C-c C-x 0	(org-timer-start &optional OFFSET)	Start or reset the relative timer. By default, the timer is set to 0. - With 'C-u' prefix: prompt the user for a starting offset. The prompt will default to a timer string at point (if any), providing a convenient way to restart taking notes after a break in the process. - With C-u C-u' prefix: change all timer strings in the active region by a certain amount. This can be used to fix timer strings if the timer was not started at exactly the right moment.									
Start a countdown timer	C-c C-x ;	(org-timer-set-timer &optional OPT)	Start a countdown timer. Prompt for a duration in minutes or hh:mm:ss. If 'org-timer-default-timer' is not "0", suggest this value as the default duration for the timer.									
	- If a timer is already set, prompt the user if she wants to replace it. - With a numeric prefix argument OPT, use this numeric value as the duration of the timer in minutes. - With 'C-u' prefix: use 'org-timer-default-timer' without prompting the user for a duration. - With C-u C-u' prefix: use 'org-timer-default-timer' without prompting the user for a duration and automatically replace any running timer. - By default, the timer duration will be set to the number of minutes in the Effort property, if any. You can ignore this with C-u C-u C-u' prefix. 👉 This command is available as ; in agenda buffers.											
Managing started timers	Once a timer is started, you can control them with the following commands.											
Insert a relative time into the buffer	C-c C-x .	(org-timer &optional RESTART NO-INSERT)	Insert a H:MM:SS string from the timer into the buffer. The first time this command is used, the timer is started.									
	- With 'C-u' prefix: force restarting the timer. - With C-u C-u' prefix: change all the timer strings in the region by a fixed amount. This can be used to re-calibrate a timer not started at the correct moment.											
Insert a description list item with current relative time	C-c C-x -	(org-timer-item &optional ARG)	Insert a description list item with the current relative time. With a prefix argument, first reset the timer to 0.									
Pause timer or continue it if already paused	C-c C-x ,	(org-timer-pause-or-continue &optional STOP)	Pause or continue the relative or countdown timer. With prefix arg STOP, stop it entirely.									
Stop the timer	C-c C-x _	(org-timer-stop)	Stop the relative or countdown timer. After this, you can only start a new timer, not continue the old one. This command also removes the timer from the mode line.									
Tables	Org tables are formatted with ASCII character, with operations that are quite flexible and powerful. - Any line with ' ' as the first non-whitespace character is considered part of a table. ' ' is also the column separator. - A line starting with ' -' is a horizontal rule. It separates rows explicitly. - Rows before the first horizontal rule are header lines. Org comes with a fast and intuitive table editor. Spreadsheet-like calculations are supported using the Emacs Calc package👉 It is possible to create cells with multiple lines in them using one of the following methods:											
Tables with multi-lines cells	Option 1: The Native Org Table with Column Wrapping using a Width Cookie (<30>)		<table><tr><th>ID</th><th>Title</th><th>Description</th></tr><tr><td>1</td><td>Short Item</td><td>This fits easily on one line.</td></tr><tr><td>2</td><td>Long Item</td><td>This is a very long description that spans multiple lines visually when you format it.</td></tr></table> <30>	ID	Title	Description	1	Short Item	This fits easily on one line.	2	Long Item	This is a very long description that spans multiple lines visually when you format it.
	ID	Title	Description									
1	Short Item	This fits easily on one line.										
2	Long Item	This is a very long description that spans multiple lines visually when you format it.										
Option 2: The table.el Format (True Multi-Line Cells)		<table><tr><th>ID</th><th>Title</th><th>Description</th></tr><tr><td>1</td><td>Short Item</td><td>This fits easily on one line.</td></tr><tr><td>2</td><td>Long Item</td><td>This is a massive block of text that splits over multiple rows seamlessly inside the plain text markup syntax.</td></tr></table>		ID	Title	Description	1	Short Item	This fits easily on one line.	2	Long Item	This is a massive block of text that splits over multiple rows seamlessly inside the plain text markup syntax.
ID	Title	Description										
1	Short Item	This fits easily on one line.										
2	Long Item	This is a massive block of text that splits over multiple rows seamlessly inside the plain text markup syntax.										
orgtbl minor mode	Although the following rows describe the commands for dealing with tables in org-mode, a major mode, you can also use them in other major modes like text-mode, mail-mode or other by activating the orgtbl minor mode.											
Toggle OrgTbl minor mode	<f11> M-o M-o	(orgtbl-mode &optional ARG)	Toggle the OrgTbl minor mode, allowing Org Table syntax handling. If the prefix argument is positive, enable the mode, and if it is zero or negative, disable the mode.									
	👉 With PEL, use the pel-mode-setup-info command bound globally to <f11> ? / (or to <f12> ? in some major modes), to display information about the current major mode and list the customizable user-options you can use to automatically activate a minor modes such as this one.											
The table.el package	Several packages cooperate with Org to extend its capabilities.											
Convert to table.el table layout	C-c -	(org-table-create-with-table.el)	Use the table.el package to insert a new table. If there is already a table at point, convert between Org tables and table.el tables.									
Edit Tables	A table is re-aligned automatically each time you press TAB, RET, or C-c C-c inside the table. - TAB also moves to the next field—RET to the next row—and creates new table rows at the end of the table or before horizontal lines. - The indentation of the table is set by the first line. Horizontal rules are automatically expanded on every re-align to span the whole table width.											
	To create the above table, type: Name Phone Age - and press TAB to align the table and start filling in fields.		Even faster; type: Name Phone Age followed by C-c RET.									
	- When typing text into a field, Org treats DEL, Backspace, and all character keys in a special way, so that inserting and deleting avoid shifting other fields. - When typing immediately after point was moved into a new field with TAB, S-TAB or RET, the field is automatically made blank. 👉 This behaviour is customizable via the org-table-auto-blank-field user-option.											
	Conversion & Creation											
Convert table from region (or create empty table if no active region)	C-C	(org-table-create-or-convert-from-region ARG)	Convert region to table, or create an empty table if there are no active region. If there is an active region, convert it to a table, using the function 'org-table-convert-region'.									
	- If every line contains at least one TAB character, the function assumes that the material is tab separated. - If every line contains a comma, comma-separated values (CSV) are assumed. - If not, lines are split at whitespace into fields. - Use a prefix argument to force a specific separator: - C-u forces CSV, - C-u C-u forces TAB, - C-u C-u C-u prompts for a regular expression to match the separator, and a numeric argument N indicates that at least N consecutive spaces, or alternatively a TAB will be the separator.											
Re-align and field motion												
Re-align	C-c C-c	(org-table-align)	Align the table at point by aligning all vertical bars.									
Re-align, go to next cell	<tab>	(org-table-next-field)	Go to the next field in the current table, creating new lines as needed. Before doing so, re-align the table if necessary.									
Re-align, go to previous cell	S-<tab>	(org-table-previous-field)	Go to the previous field in the table. Before doing so, re-align the table if necessary.									

Description	Keystroke	Function	Note
Re-align, go down	RET	(org-table-next-row)	Go to the next row (same column) in the current table. When next row is an hline or outside the table, create a new empty row. Before doing so, re-align the table if necessary.
Go to beginning of field or previous	M-a	(org-table-beginning-of-field &optional N)	Move to the beginning of the current table field. If already at or before the beginning, move to the beginning of the previous field. With numeric argument N, move N-1 fields backward first.
Go to end of field or next	M-e	(org-table-end-of-field &optional N)	Move to the end of the current table field. If already at or after the end, move to the end of the next table field. With numeric argument N, move N-1 fields forward first.
Blank current cell	<f12> t <SPC>	(org-table-blank-field)	Blank the current table field or active region.
• Column & Row Editing			
Move column to the left	M-<left>	(org-table-move-column-left)	Move column to the left.
Move column to the right	M-<right>	(org-table-move-column-right)	Move column to the right.
Move row up	M-<up>	(org-table-move-row-up)	Move table row up.
Move row down	M-<down>	(org-table-move-row-down)	Move table row down.
Move current cell to the left	S-<left>	(org-table-move-cell-left)	Move a single cell left in a table. Swap with anything in target cell.
Move current cell to the right	S-<right>	(org-table-move-cell-right)	Move a single cell right in a table. Swap with anything in target cell.
Move current cell up	S-<up>	(org-table-move-cell-up)	Move a single cell up in a table. Swap with anything in target cell.
Move current cell down	S-<down>	(org-table-move-cell-down)	Move a single cell down in a table. Swap with anything in target cell.
Delete column	M-S-<left>	(org-table-delete-column)	Delete current column from the table.
Kill current row	M-S-<up>	(org-table-kill-row)	Kill the current row or horizontal line from the table. Stored in the kill ring.
Insert horizontal line	C-c -	(org-table-insert-hline &optional ABOVE)	Insert a horizontal-line below the current line into the table. With prefix ABOVE, insert above the current line.
Insert horizontal line, move down	C-c RET	(org-table-hline-and-move &optional SAME-COLUMN)	Insert a hline and move to the row below that line.
Insert a column	M-S-<right>	(org-table-insert-column)	Insert a new column into the table before (to the left of) the current column.
Insert a row	M-S-<down>	(org-table-insert-row &optional ARG)	Insert a new row above the current line into the table. With prefix ARG, insert below the current line.
Sort the table lines	C-C ^	(org-table-sort-lines &optional WITH-CASE SORTING-TYPE GETKEY-FUNC COMPARE-FUNC INTERACTIVE?)	Sort the table lines in the region. - The position of point indicates the column to be used for sorting, and the range of lines is the range between the nearest horizontal separator lines, or the entire table. - If point is before the first column, you are prompted for the sorting column. - If there is an active region, the mark specifies the first line and the sorting column, while point should be in the last line to be included into the sorting. - The command prompts for the sorting type, alphabetically, numerically, or by time. - You can sort in normal or reverse order. - You can also supply your own key extraction and comparison functions. - When called with a prefix argument, alphabetic sorting is case-sensitive.
• Regions			
Copy rectangle region	C-c C-x M-w	(org-table-copy-region BEG END &optional CUT)	Copy rectangular region in table to clipboard. - A special clipboard is used which can only be accessed with ‘org-table-paste-rectangle’. - Return the region copied, as a list of lists of fields.
Cut rectangle region	C-c C-x C-w	(org-table-cut-region BEG END)	Copy region in table to the clipboard and blank all relevant fields. If there is no active region, use just the field at point
Paste a rectangle	C-c C-x C-y	(org-table-paste-rectangle)	Paste a rectangular region into a table. The upper right corner ends up in the current field. All involved fields will be overwritten. If the rectangle does not fit into the present table, the table is enlarged as needed. The process ignores horizontal separator lines.
Wrap cell in multiple lines	M-RET	(org-table-wrap-region ARG)	Wrap several fields in a column like a paragraph. - This is useful if you’d like to spread the contents of a field over several lines, in order to keep the table compact. However, each new line is a new table row. - If there is an active region, and both point and mark are in the same column, the text in the column is wrapped to minimum width for the given number of lines. A numeric prefix argument may be used to change the number of desired lines. If there is no region, but you specify a prefix argument, the current field is made blank, and the content is appended to the field above.
• Calculations			
Sum table rows	C-c +	(org-table-sum &optional BEG END NLAST)	Sum the numbers in the current column, or in the rectangle defined by the active region. The result is shown in the echo area and the text can be inserted with C-y (but inserted normally which push the cell alignment if you yank it inside a table cell).
Copy above field in current one • increment integer value	• S-RET • <f12> t RET	(org-table-copy-down N)	When the current field is empty, copy from the first non-empty field above. - This is useful if you’d like to spread the contents of a field over several lines, in order to keep the table compact. However, each new line is a new table row. - Depending on the variable org-table-copy-increment, integer and time stamp field values, as well as fields prefixed or suffixed with a whole number, can be incremented during copy. - Also, a 0 prefix argument temporarily disables the increment. - This key is also used by shift-selection and related modes (see Packages that conflict with Org mode).
• Miscellaneous			
Edit table field in different window	C-c `	(org-table-edit-field ARG)	Edit table field in a different window. This is mainly useful for fields that contain hidden parts. - When called with a ‘C-u’ prefix, just make the full field visible so that it can be edited in place. - When called with a ‘C-u C-u’ prefix, toggle ‘org-table-follow-field-mode’.
Import file as table	<f12> t i	(org-table-import FILE SEPARATOR)	Import FILE as a table. The command tries to be smart and figure out the separator in the following way: - when each line contains a TAB, assume TAB-separated material; - when each line contains a comma, assume CSV material; - else, assume one or more SPACE characters as separator. When non-nil, SEPARATOR specifies the field separator in the lines. It can have the following values: (4) Use the comma as a field separator. (16) Use a TAB as field separator. (64) Prompt for a regular expression as field separator. - integer When a number, use that many spaces, or a TAB, as field separator. - regexp When a regular expression, use it to match the separator.
Toggle table header minor mode: show table title at top of buffer if it is not visible	<f12> t t	(org-table-header-line-mode &optional ARG)	Toggle the org-table-header-line-mode, a minor mode. This mode display the first row of the table at point in the header line.

Description	Keystroke	Function	Note																												
Transpose Org Table	<f12> t M-t	(org-table-transpose-table-at-point)	Transpose Org table at point and eliminate hlines. So a table like <table><tr><td>1</td><td>2</td><td>4</td><td>5</td></tr><tr><td>---</td><td>---</td><td>---</td><td>---</td></tr><tr><td>a</td><td>b</td><td>c</td><td>d</td></tr><tr><td>e</td><td>f</td><td>g</td><td>h</td></tr></table> will be transposed as <table><tr><td>1</td><td>a</td><td>e</td></tr><tr><td>2</td><td>b</td><td>f</td></tr><tr><td>4</td><td>c</td><td>g</td></tr><tr><td>5</td><td>d</td><td>h</td></tr></table> Note that horizontal lines disappear.	1	2	4	5	---	---	---	---	a	b	c	d	e	f	g	h	1	a	e	2	b	f	4	c	g	5	d	h
1	2	4	5																												
---	---	---	---																												
a	b	c	d																												
e	f	g	h																												
1	a	e																													
2	b	f																													
4	c	g																													
5	d	h																													
Spreadsheet	Emacs Org mode includes a powerful built-in spreadsheet system. It lets you run math formulas, use column and row references, calculate table data automatically, and export results seamlessly inside plain text files. Key features - Formula Editor: Press C-c ' to open a separate window to write and edit formulas easily. - Cell References: Use Excel-style coordinates like \$1, \$2 for columns, or @2\$3 for specific row and column intersections. - Ranges and Math: Compute sums, averages, and standard formulas like vsum(\$1..\$3) or standard arithmetic (\$1 + \$2). - Automatic Recalculation: Press C-c C-c on a formula line or table field to update all values instantly. - Field Formatting: Format numbers as currency, integers, or decimals using standard format specifiers (e.g., %.2f). Advanced Uses - Named Fields: Assign names to specific cells or ranges to make formulas easier to read. - Elisp Integration: Run custom Emacs Lisp code inside formulas for advanced text or data processing. - TBLFM Lines: Store formulas at the bottom of your tables so they stay saved with your document. By default, the Org Spreadsheet uses the GNU Emacs Calc as its primary math engine. It also supports algebra equation resolution as does calc. You can also use Emacs calc independently of Org mode: start Emacs calc with C-x * c It will open 2 windows to interact with the calculator.																														
Info on Calc in Emacs	C-h i m calc	Once inside the Info manual you can type T to get a tree of all nodes in the manual. You can then quickly search the section.																													
How to enter formula in cells	Formulas can be entered inside cells: - with the #+TBLFM: keyword (see below) - in-cell column formula, applying a formula to an entire column by typing = followed by the formula inside one cell of the column. Like: =\$1+\$2 <tab> - in-cell single field formula: calculate only a specific cell value. Like: :=vsum(@2..@-1) <tab> - via interactive prompt with C-c = to type column formula or C-u C-c = for single field formula - via a dedicated formula Editor: C-c '																														
Example of TBLFM - sum	<table><tr><td>Item</td><td>Cost</td></tr><tr><td>-----+-----</td><td></td></tr><tr><td>Coffee</td><td>4.50</td></tr><tr><td>Lunch</td><td>21.00</td></tr><tr><td>Book</td><td>29.95</td></tr><tr><td>-----+-----</td><td></td></tr><tr><td>Total</td><td>⊕</td></tr></table> #+TBLFM: @5\$2=vsum(@2\$2..@4\$2);%.2f	Item	Cost	-----+-----		Coffee	4.50	Lunch	21.00	Book	29.95	-----+-----		Total	⊕	#+TBLFM: Tells Org mode that this line contains the table formulas. @5\$2 Targets the cell at Row 5, Column 2 (the Total amount). - vsum(@2\$2..@4\$2) Adds up all the values from Row 2, Column 2 down to Row 4, Column 2. %.2f Formats the resulting number to show exactly two decimal places. To compute the formula, place point over #+TBLFM: and press C-c C-c. The sum value, 55.45, will be computed and placed in the Total row second column, where ⊕ shows up now.															
Item	Cost																														
-----+-----																															
Coffee	4.50																														
Lunch	21.00																														
Book	29.95																														
-----+-----																															
Total	⊕																														
Example of TBLFM - Algebra Equation Resolution	<table><tr><td>Variable</td><td>Equation</td><td>Result</td></tr><tr><td>-----+-----+-----</td><td></td><td></td></tr><tr><td>x</td><td>2*x + 5 = 15</td><td>⊕</td></tr></table> #+TBLFM: \$3=solve(\$2, \$1)	Variable	Equation	Result	-----+-----+-----			x	2*x + 5 = 15	⊕	This uses the solve(eq, var) where the variable in \$1, x, is unknown, and the equation \$2 is 2*x + 5 = 15. The solution is 5, placed where ⊕ shows up now.																				
Variable	Equation	Result																													
-----+-----+-----																															
x	2*x + 5 = 15	⊕																													
Example of TBLFM - Algebra: a 2-variables Equation Resolution	The system also supports solving multiple equations, just as calc does. The result cell, after execution is shown in the example.	<table><tr><td>Equations</td><td>Variables</td><td>Result</td></tr><tr><td>-----+-----+-----</td><td></td><td></td></tr><tr><td>[x + y = 10, x - y = 8]</td><td>[x, y]</td><td>[x = 9, y = 1]</td></tr></table> #+TBLFM: \$3=solve(\$1, \$2)	Equations	Variables	Result	-----+-----+-----			[x + y = 10, x - y = 8]	[x, y]	[x = 9, y = 1]																				
Equations	Variables	Result																													
-----+-----+-----																															
[x + y = 10, x - y = 8]	[x, y]	[x = 9, y = 1]																													
Example of TBLFM - Algebra: a 2-variables Equation Resolution - multi-cell splitting.	If you prefer to keep your equations grouped but want the final results isolated side-by-side, you can reference the solved vector cell directly using standard cell coordinates.	<table><tr><td>Equation 1</td><td>Equation 2</td><td>x</td><td>y</td></tr><tr><td>-----+-----+-----</td><td></td><td></td><td></td></tr><tr><td>x + y = 10</td><td>x - y = 8</td><td>9</td><td>1</td></tr></table> #+TBLFM: \$3=vsubst(solve([\$1, \$2], [x, y]), x)[1];\ #+TBLFM: \$4=vsubst(solve([\$1, \$2], [x, y]), y)[2]	Equation 1	Equation 2	x	y	-----+-----+-----				x + y = 10	x - y = 8	9	1																	
Equation 1	Equation 2	x	y																												
-----+-----+-----																															
x + y = 10	x - y = 8	9	1																												
Example of TBLFM - Symbolic Differentiation Equation	<table><tr><td>Function (f(x))</td><td>Derivative (f'(x))</td></tr><tr><td>-----+-----</td><td></td></tr><tr><td>x^3 + 2*x</td><td>3 * x^2 + 2</td></tr><tr><td>sin(x)</td><td>cos(x)</td></tr><tr><td>ln(x)</td><td>1 / x</td></tr></table> #+TBLFM: \$2=deriv(\$1, x)	Function (f(x))	Derivative (f'(x))	-----+-----		x^3 + 2*x	3 * x^2 + 2	sin(x)	cos(x)	ln(x)	1 / x	In ☞, Org mode passes the mathematical text from Column 1 into deriv(), tells it to differentiate with respect to the variable x, and Calc outputs the symbolic calculus solution directly into Column 2.																			
Function (f(x))	Derivative (f'(x))																														
-----+-----																															
x^3 + 2*x	3 * x^2 + 2																														
sin(x)	cos(x)																														
ln(x)	1 / x																														
Example of TBLFM - Finding the Slopes	<table><tr><td>Point (x)</td><td>Function</td><td>Slope (f'(x))</td></tr><tr><td>-----+-----+-----</td><td></td><td></td></tr><tr><td>3</td><td>x^2</td><td>6</td></tr><tr><td>5</td><td>x^2</td><td>10</td></tr></table> #+TBLFM: \$3=subst(deriv(\$2, x), x, \$1)	Point (x)	Function	Slope (f'(x))	-----+-----+-----			3	x^2	6	5	x^2	10	\textbf{Quadratic Function and its Derivative}																	
Point (x)	Function	Slope (f'(x))																													
-----+-----+-----																															
3	x^2	6																													
5	x^2	10																													
Plot curves from their data with Gnuplot as show here: The #+PLOT must be one a single line	#+PLOT: title:"Function and Derivative" ind:1 deps:(2 3) type:2d with:lines file:"plot.png"																														
	<table><tr><td>Point (x)</td><td>Function (x^2)</td><td>Slope (2x)</td></tr><tr><td>-----+-----+-----</td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>2</td></tr><tr><td>2</td><td>4</td><td>4</td></tr><tr><td>3</td><td>9</td><td>6</td></tr><tr><td>4</td><td>16</td><td>8</td></tr></table>	Point (x)	Function (x^2)	Slope (2x)	-----+-----+-----			1	1	2	2	4	4	3	9	6	4	16	8												
Point (x)	Function (x^2)	Slope (2x)																													
-----+-----+-----																															
1	1	2																													
2	4	4																													
3	9	6																													
4	16	8																													
Use Org Babel to write code to generate the curve from its formula. It can use any programming language but can also use the mathematical functions like this:	#+begin_src gnuplot :file equation_plot.png set title "Function and its Derivative" set xlabel "X Axis" set ylabel "Y Axis" set grid set xrange [-5:5] # Define/plot the continuous math. equations plot x**2 title "f(x) = x^2" with lines, \ 2*x title "f'(x) = 2x" with lines #+end_src																														
Use complex numbers in formulas to perform engineering computations!	Use calc inside a Org spreadsheet to compute the total impedance Ztotal Of a parallel AC circuit.	$Z_{total} = \frac{1}{\frac{1}{R} + \frac{1}{jX_L}}$	To do that, write the formula inside a #+TBLFM: expression at the end of the table. ☺ Note the j at the end of the formula to tell calc to interpret the input as complex numbers !!																												
		$R_1 = 10 \, \Omega$$X_L = 15 \, \Omega$$\text{Impedance of } R_1(Z_R) = 10 + j0 \, \Omega$$\text{Impedance of } L_1(Z_L) = 0 + j15 \, \Omega$$Z_{Total} = \left(\frac{1}{Z_R} + \frac{1}{Z_L} \right)^{-1}$$Z_{Total} = \left(\frac{1}{10 + j0} + \frac{1}{0 + j15} \right)^{-1}$$Z_{Total} \approx 8.32 + j5.55 \, \Omega$	<table><tr><td>Component</td><td>Resistance (R)</td><td>Reactance (X)</td><td>Impedance (Z)</td></tr><tr><td>-----+-----+-----</td><td></td><td></td><td></td></tr><tr><td>R1</td><td>10</td><td>0</td><td>(10, 0)</td></tr><tr><td>L1</td><td>0</td><td>15</td><td>(0, 15)</td></tr><tr><td>-----+-----+-----</td><td></td><td></td><td></td></tr><tr><td>Total</td><td></td><td></td><td>(8.32, 5.547)</td></tr></table> #+TBLFM: @2\$4=(\$2, \$3);c :: @3\$4=(\$2, \$3);c :: @4\$4=1 / ((1 / @2\$4) + (1 / @3\$4));c	Component	Resistance (R)	Reactance (X)	Impedance (Z)	-----+-----+-----				R1	10	0	(10, 0)	L1	0	15	(0, 15)	-----+-----+-----				Total			(8.32, 5.547)				
Component	Resistance (R)	Reactance (X)	Impedance (Z)																												
-----+-----+-----																															
R1	10	0	(10, 0)																												
L1	0	15	(0, 15)																												
-----+-----+-----																															
Total			(8.32, 5.547)																												

Description	Keystroke	Function	Note
Editing and Debugging Formulas	- While editing a formula: 🧐 you may want to turn off Org-mode electric pairing of markup characters used in formulas. Use <code><f11> M-e</code> for that (1). - To debug: when the evaluation of a formula leads to an error, the field content becomes the string <code>'#ERROR'</code>. - To see what is going on during variable substitution and calculation in order to find a bug, turn on formula debugging with <code>C-c {</code> in the table and repeat the calculation, for example by pressing <code>C-u C-u C-c = RET</code> in a field. Detailed information is displayed.		
Install a new formula in the cell	<code>C-c =</code>	(org-table-eval-formula &optional ARG EQUATION SUPPRESS-ALIGN SUPPRESS-CONST SUPPRESS-STORE SUPPRESS-ANALYSIS)	Inside a table cell: Install a new formula for the current column and replace current field with the result of the formula. - The command prompts for a formula, with default taken from the <code>'TBLFM'</code> keyword, applies it to the current field and stores it. - With a numeric prefix argument, e.g., <code>C-5 C-c =</code>, the command applies it to that many consecutive fields in the current column.
Recalculate formulas inside current table	<code>C-c *</code>	(org-table-recalculate &optional ALL NOALIGN)	Inside a table cell: Recalculate the current table line by applying all stored formulas. - With prefix arg <code>ALL</code>, do this for all lines in the table. - With a <code>'C-u C-u'</code> prefix, or if <code>ALL</code> is the symbol 'iterate', recompute the table until it no longer changes. If <code>NOALIGN</code> is not nil, do not re-align the table after the computations are done. This is typically used internally to save time, if it is known that the table will be realigned a little later anyway.
Recalculate formulas inside all tables in current buffer	<code><f12> t =</code>	(org-table-recalculate-buffer-tables)	Recalculate all tables in the current buffer.
Iterate recalculate formulas inside all tables in current buffer to converge inter-table dependencies	<code><f12> t M-=</code>	(org-table-iterate-buffer-tables)	Iterate all tables in the buffer, in order to converge inter-table dependencies.
💡 Generate all reports in a master time tracking file		🧐 A formula inside a table can refer to another table. There are 2 possible syntaxes: - Name the Source Table: Use <code>#+NAME: table_a</code> directly above your source data. - Reference it in the Destination Table: Use <code>remote(table_a, ...)</code> inside the <code>#+TBLFM:</code> formula string. 💡 Use this to create or refresh the time tracking reports in master project time tracking file. - See the PEL example/templates/org-mode/master-org.org file for an example.	
Rotate calculation mark in first column	<code>C-#</code> <code><f12> t #</code>	(org-table-rotate-recalc-marks &optional NEWCHAR)	Rotate the recalculation mark in the first column. - If in any row, the first field is not consistent with a mark, insert a new column for the markers. - When there is an active region, change all the lines in the region, after prompting for the marking character. - After each change, a message will be displayed indicating the meaning of the new mark.
Show formula info for cell	<code>C-c ?</code>	(org-table-field-info ARG)	Inside a table cell: Show info about the current field, and highlight any reference at point.
Toggle display of Row/Column overlay	<code>C-c }</code>	(org-table-toggle-coordinate-overlays)	Inside a table cell: Toggle the display of Row/Column numbers in tables.
Toggle formula debugger	<code>C-c {</code>	(org-table-toggle-formula-debugger)	Inside a table cell: Toggle the formula debugger in tables.
Edit the formulas in formula editor buffer	<code>C-c '</code>	(org-table-edit-formulas)	Inside a table cell: Edit the formulas of the current table in a separate buffer.
Edit cell in other window	<code>C-c `</code>	(org-table-edit-field ARG)	Inside a table cell: Edit table field in a different window. When done, finish with <code>C-c C-c</code> - This is mainly useful for fields that contain hidden parts. - With a <code>'C-u'</code> prefix: just make the full field visible so that it can be edited in place. - With a <code>'C-u C-u'</code> prefix: toggle 'org-table-follow-field-mode'.
Toggle reference format	<code>C-c C-r</code>	(org-table-fedit-toggle-ref-type)	Inside "Edit Formula" buffer: Convert all references in the buffer from <code>B3</code> to <code>@3\$2</code> and back. This converts the reference formats between the Org-style (like <code>'@3\$2'</code> or <code>'\$4'</code>) to more conventional spreadsheet format (like <code>'B3'</code> or <code>'D&')</code>.
Pretty print elisp formula	<code><tab></code>	(org-table-fedit-lisp-indent)	Inside "Edit Formula" buffer: Pretty-print and re-indent Lisp expressions in the Formula Editor.
Tab completion of elisp function	<code>M-<tab></code>	(lisp-complete-symbol)	Inside "Edit Formula" buffer: Complete Lisp symbols, just like in Emacs Lisp mode.
Shift the reference 1 row up	<code>S-<up></code>	(org-table-fedit-ref-up)	Inside "Edit Formula" buffer when point over a reference: Shift the reference at point one row/hline up.
Shift the reference 1 row down	<code>S-<down></code>	(org-table-fedit-ref-down)	Inside "Edit Formula" buffer when point over a reference: Shift the reference at point one row/hline down.
Shift the reference 1 row left	<code>S-<left></code>	(org-table-fedit-ref-left)	Inside "Edit Formula" buffer when point over a reference: Shift the reference at point one field to the left.
Shift the reference 1 row right	<code>S-<right></code>	(org-table-fedit-ref-right)	Inside "Edit Formula" buffer when point over a reference: Shift the reference at point one field to the right.
Move table cursor up	<code>M-S-<up></code>	(org-table-fedit-line-up)	Inside "Edit Formula" buffer: Move cursor one line up in the window showing the table.
Move table cursor down	<code>M-S-<down></code>	(org-table-fedit-line-down)	Inside "Edit Formula" buffer: Move cursor one line down in the window showing the table.
Scroll table window up	<code>M-<up></code>	(org-table-fedit-scroll-down N)	Inside "Edit Formula" buffer: Scroll up the window displaying the table.
Scroll table window down	<code>M-<down></code>	(org-table-fedit-scroll-down N)	Inside "Edit Formula" buffer: Scroll down the window displaying the table.
Toggle display of coordinates	<code>C-c }</code>	(org-table-fedit-toggle-coordinates)	Inside "Edit Formula" buffer: Toggle the display of coordinates in the referenced table.
Complete formula editing	<code>C-c C-c C-x C-s</code>	(org-table-fedit-finish &optional ARG)	Inside "Edit Formula" buffer: Parse the buffer for formula definitions and install them. With prefix <code>ARG</code>, apply the new formulas to the table.
Abort formula editing	<code>C-c C-q</code>	(org-table-fedit-abort)	Inside "Edit Formula" buffer: Abort editing formulas, without installing the changes.

Org-Mode - Reference

Topic & link	Description
Org Mode @ Wikipedia	A good introduction to Org Mode concepts
🔗 Org mode home and 🔗 The Org Manual	Org mode is actively developed. The Org Mode home page contains links to all you need.
Org Mode CHANGES and previous releases	
Org-mode Contributed Packages	A list of contributed packages that extend Org mode. Some have links to the files.
David 'Otool Org Tutorial	Teaches basic org-mode usage with step-by-step instructions and screenshots. Good starting point.
Reddit r/orgmode	A Reddit channel on Org Mode.
Quick Reference for emacs Org-Mode key binds	Quick overview of basic commands organized by feature. Nice to start but incomplete.
Getting started with Org-Mode @ Youtube	
Org Mode features You May Not Know	By Bastien Guerry
How to move a subtree to another subtree in org-mode emacs	Stack Overflow Q&A. Just to move the sub-tree itself (not its top level headline).
Command to copy and delete subtree in org-mode	Stack Overflow Q&A. Answer: C-C C-x C-w and C-c C-x C-y
Literate DevOps with Emacs - youtube video	
Hello Worg, the Org-Mode Community!	
Org Mode Tools	
Org-Mode Exporters - The New Org-mode Exporter Framework	
* ox.el @ github	
Pandoc support	
Pandoc-mode - An Emacs mode for interacting with Pandoc	
Org mode and Pandoc	
Keeping a lab notebook with org-mode, git, Papers, and Pandoc (Part II)	
Confluence Wiki Support for Org Mode	• confluence-el (10 years old) , confluence-el : seems newer. • ox-confluence @ GitHub • In org-mode, how do I export to JIRA wiki markup? @ StackExchange
Clocking Work Time - The Org Manual	
• Time Tracking in Emacs with org-clock	Short article written by David Charte, on November 2017.
• Clocking time with Org-mode	Good description of the clocking time mechanism supported by org-mode. Written in April 2017 by Lee Hinman
• Org Mode Time and Task Tools – Straightforward Emacs @YouTube	
• Emacs Org id will change how you see things @YouTube	From Chris Maiorama
• Org Mode Citation and Footnote Features — Straightforward Emacs	Jake B YouTube video presenting Emacs org-mode citation and footnote features.
• Writing customized clock table formatters	Writing customized Org mode clock table formatter is possible in Emacs Lisp. It is however not very well documented (or I have not found it yet). I have seen the following: • How to choose the order of clocktable columns? on Emacs StackExchange • Is it really this complicated to customize a clock report? on Reddit r/orgmode
The GNU Emacs Calculator	That's a link to an old version of calc manual that is still valid (but incomplete). Use Emacs info to read the manual.
• The Emacs Calculator - Chris Mellon, June 2009.	The GNU Emacs Calculator is more than what its name says. It's really a computer algebra system.