

Org-Mode Reference Card (1/2)

(for version 9.1.9)

Getting Started

To read the on-line documentation try `M-x org-info`

Visibility Cycling

rotate current subtree between states	<code>TAB</code>
rotate entire buffer between states	<code>S-TAB</code>
restore property-dependent startup visibility	<code>C-u C-u TAB</code>
show the whole file, including drawers	<code>C-u C-u TAB</code>
reveal context around point	<code>C-c C-r</code>

Motion

next/previous heading	<code>C-c C-n/p</code>
next/previous heading, same level	<code>C-c C-f/b</code>
backward to higher level heading	<code>C-c C-u</code>
jump to another place in document	<code>C-c C-j</code>
previous/next plain list item	<code>S-UP/DOWN</code> [2]

Structure Editing

insert new heading/item at current level	<code>M-RET</code>
insert new heading after subtree	<code>C-RET</code>
insert new TODO entry/checkbox item	<code>M-S-RET</code>
insert TODO entry/checkbox after subtree	<code>C-S-RET</code>
turn (head)line into item, cycle item type	<code>C-c -</code>
turn item/line into headline	<code>C-c *</code>
promote/demote heading	<code>M-LEFT/RIGHT</code>
promote/demote current subtree	<code>M-S-LEFT/RIGHT</code>
move subtree/list item up/down	<code>M-UP/DOWN</code>
move the line at point up/down	<code>M-S-UP/DOWN</code>
sort subtree/region/plain-list	<code>C-c ^</code>
clone a subtree	<code>C-c C-x c</code>
copy visible parts of the region	<code>C-c C-x v</code>
kill/copy subtree	<code>C-c C-x C-w/M-w</code>
yank subtree	<code>C-c C-x C-y</code> or <code>C-y</code>
narrow buffer to subtree / widen	<code>C-x n s/w</code>

Capture - Refile - Archiving

capture a new item (<code>C-u C-u = goto last</code>)	<code>C-c c</code> [1]
refile subtree (<code>C-u C-u = goto last</code>)	<code>C-c C-w</code>
archive subtree using the default command	<code>C-c C-x C-a</code>
move subtree to archive file	<code>C-c C-x C-s</code>
toggle ARCHIVE tag / to ARCHIVE sibling	<code>C-c C-x a/A</code>
force cycling of an ARCHIVED tree	<code>C-TAB</code>

Filtering and Sparse Trees

construct a sparse tree by various criteria	<code>C-c /</code>
view TODO's in sparse tree	<code>C-c / t/T</code>
global TODO list in agenda mode	<code>C-c a t</code> [1]

Tables

Creating a table

just start typing, e.g. `|Name|Phone|Age RET |- TAB`
 convert region to table `C-c |`
 ... separator at least 3 spaces `C-3 C-c |`

Commands available inside tables

The following commands work when the cursor is *inside a table*. Outside of tables, the same keys may have other functionality.

Re-aligning and field motion

re-align the table without moving the cursor	<code>C-c C-c</code>
re-align the table, move to next field	<code>TAB</code>
move to previous field	<code>S-TAB</code>
re-align the table, move to next row	<code>RET</code>
move to beginning/end of field	<code>M-a/e</code>

Row and column editing

move the current column left	<code>M-LEFT/RIGHT</code>
kill the current column	<code>M-S-LEFT</code>
insert new column to left of cursor position	<code>M-S-RIGHT</code>
move the current row up/down	<code>M-UP/DOWN</code>
kill the current row or horizontal line	<code>M-S-UP</code>
insert new row above the current row	<code>M-S-DOWN</code>
insert hline below (<code>C-u</code> : above) current row	<code>C-c -</code>
insert hline and move to line below it	<code>C-c RET</code>
sort lines in region	<code>C-c ^</code>

Regions

cut/copy/paste rectangular region `C-c C-x C-w/M-w/C-y`

Miscellaneous

to limit column width to N characters, use	<code>... <N> ...</code>
edit the current field in a separate window	<code>C-c '</code>
make current field fully visible	<code>C-u TAB</code>
export as tab-separated file	<code>M-x org-table-export</code>
import tab-separated file	<code>M-x org-table-import</code>
sum numbers in current column/rectangle	<code>C-c +</code>

Tables created with the table.el package

insert a new <code>table.el</code> table	<code>C-c ~</code>
recognize existing <code>table.el</code> table	<code>C-c C-c</code>
convert table (Org-mode ↔ <code>table.el</code>)	<code>C-c ~</code>

Spreadsheet

Formulas typed in field are executed by `TAB`, `RET` and `C-c C-c`. = introduces a column formula, := a field formula.

Example: Add Col1 and Col2	<code> =\$1+\$2 </code>
... with printf format specification	<code> =\$1+\$2;%.2f </code>
... with constants from constants.el	<code> =\$1/\$c/\$cm </code>
sum from 2nd to 3rd hline	<code> :=vsum(@II..@III) </code>
apply current column formula	<code> = </code>
set and eval column formula	<code>C-c =</code>
set and eval field formula	<code>C-u C-c =</code>
re-apply all stored equations to current line	<code>C-c *</code>
re-apply all stored equations to entire table	<code>C-u C-c *</code>
iterate table to stability	<code>C-u C-u C-c *</code>
rotate calculation mark through # * ! ^ _ \$	<code>C-#</code>
show line, column, formula reference	<code>C-c ?</code>
toggle grid / debugger	<code>C-c }/{</code>

Formula Editor

edit formulas in separate buffer	<code>C-c '</code>
exit and install new formulas	<code>C-c C-c</code>
exit, install, and apply new formulas	<code>C-u C-c C-c</code>
abort	<code>C-c C-q</code>
toggle reference style	<code>C-c C-r</code>
pretty-print Lisp formula	<code>TAB</code>
complete Lisp symbol	<code>M-TAB</code>
shift reference point	<code>S-cursor</code>
shift test line for column references	<code>M-up/down</code>
scroll the window showing the table	<code>M-S-up/down</code>
toggle table coordinate grid	<code>C-c }</code>

Links

globally store link to the current location	<code>C-c l</code> [1]
insert a link (<code>TAB</code> completes stored links)	<code>C-c C-l</code>
insert file link with file name completion	<code>C-u C-c C-l</code>
edit (also hidden part of) link at point	<code>C-c C-l</code>
open file links in emacs	<code>C-c C-o</code>
...force open in emacs/other window	<code>C-u C-c C-o</code>
open link at point	<code>mouse-1/2</code>
...force open in emacs/other window	<code>mouse-3</code>
record a position in mark ring	<code>C-c %</code>
jump back to last followed link(s)	<code>C-c &</code>
find next link	<code>C-c C-x C-n</code>
find previous link	<code>C-c C-x C-p</code>
edit code snippet of file at point	<code>C-c '</code>
toggle inline display of linked images	<code>C-c C-x C-v</code>

Working with Code (Babel)

execute code block at point	<code>C-c C-c</code>
open results of code block at point	<code>C-c C-o</code>
check code block at point for errors	<code>C-c C-v c</code>
insert a header argument with completion	<code>C-c C-v j</code>
view expanded body of code block at point	<code>C-c C-v v</code>
view information about code block at point	<code>C-c C-v I</code>
go to named code block	<code>C-c C-v g</code>
go to named result	<code>C-c C-v r</code>
go to the head of the current code block	<code>C-c C-v u</code>
go to the next code block	<code>C-c C-v n</code>
go to the previous code block	<code>C-c C-v p</code>
demarcate a code block	<code>C-c C-v d</code>
execute the next key sequence in the code edit buffer	<code>C-c C-v x</code>
execute all code blocks in current buffer	<code>C-c C-v b</code>
execute all code blocks in current subtree	<code>C-c C-v s</code>
tangle code blocks in current file	<code>C-c C-v t</code>
tangle code blocks in supplied file	<code>C-c C-v f</code>
ingest all code blocks in supplied file into the Library of Babel	<code>C-c C-v i</code>
switch to the session of the current code block	<code>C-c C-v z</code>
load the current code block into a session	<code>C-c C-v l</code>
view sha1 hash of the current code block	<code>C-c C-v a</code>

Completion

In-buffer completion completes TODO keywords at headline start, TeX macros after “\”, option keywords after “#-”, TAGS after “:”, and dictionary words elsewhere.

complete word at point `M-TAB`

Org-Mode Reference Card (2/2)

(for version 9.1.9)

TODO Items and Checkboxes

rotate the state of the current item	C-c C-t
select next/previous state	S-LEFT/RIGHT
select next/previous set	C-S-LEFT/RIGHT
toggle ORDERED property	C-c C-x o
view TODO items in a sparse tree	C-c / t
view 3rd TODO keyword's sparse tree	C-3 C-c / t
set the priority of the current item	C-c , [ABC]
remove priority cookie from current item	C-c , SPC
raise/lower priority of current item	S-UP/DOWN [2]
insert new checkbox item in plain list	M-S-RET
toggle checkbox(es) in region/entry/at point	C-c C-x C-b
toggle checkbox at point	C-c C-c
update checkbox statistics (C-u : whole file)	C-c #

Tags

set tags for current heading	C-c C-q
realign tags in all headings	C-u C-c C-q
create sparse tree with matching tags	C-c \
globally (agenda) match tags at cursor	C-c C-o

Properties and Column View

set property/effort	C-c C-x p/e
special commands in property lines	C-c C-c
next/previous allowed value	S-left/right
turn on column view	C-c C-x C-c
capture columns view in dynamic block	C-c C-x i
quit column view	q
show full value	v
edit value	e
next/previous allowed value	n/p or S-left/right
edit allowed values list	a
make column wider/narrower	> / <
move column left/right	M-left/right
add new column	M-S-right
Delete current column	M-S-left

Timestamps

prompt for date and insert timestamp	C-c .
like C-c . but insert date and time format	C-u C-c .
like C-c . but make stamp inactive	C-c !
insert DEADLINE timestamp	C-c C-d
insert SCHEDULED timestamp	C-c C-s
create sparse tree with all deadlines due	C-c / d
the time between 2 dates in a time range	C-c C-y
change timestamp at cursor ±1 day	S-RIGHT/LEFT [2]
change year/month/day at cursor by ±1	S-UP/DOWN [2]
access the calendar for the current date	C-c >
insert timestamp matching date in calendar	C-c <
access agenda for current date	C-c C-o
select date while prompted	mouse-1/RET
toggle custom format display for dates/times	C-c C-x C-t

Clocking time

start clock on current item	C-c C-x C-i
stop/cancel clock on current item	C-c C-x C-o/x
display total subtree times	C-c C-x C-d
remove displayed times	C-c C-c
insert/update table with clock report	C-c C-x C-r

Agenda Views

add/move current file to front of agenda	C-c [
remove current file from your agenda	C-c]
cycle through agenda file list	C-'
set/remove restriction lock	C-c C-x </>
compile agenda for the current week	C-c a a [1]
compile global TODO list	C-c a t [1]
compile TODO list for specific keyword	C-c a T [1]
match tags, TODO kwds, properties	C-c a m [1]
match only in TODO entries	C-c a M [1]
find stuck projects	C-c a # [1]
configure custom commands	C-c a C [1]
agenda for date at cursor	C-c C-o

Commands available in an agenda buffer

View Org file

show original location of item	SPC/mouse-3
show and recenter window	L
goto original location in other window	TAB/mouse-2
goto original location, delete other windows	RET
show subtree in indirect buffer, ded. frame	C-c C-x b
toggle follow-mode	F

Change display

delete other windows	o
view mode dispatcher	v
switch to day/week/month/year/def view	d w vm vy vSP
toggle diary entries / time grid / habits	D / G / K
toggle entry text / clock report	E / R
toggle display of logbook entries	l / v 1/L/c
toggle inclusion of archived trees/files	v a/A
refresh agenda buffer with any changes	r / g
filter with respect to a tag	/
save all org-mode buffers	s
display next/previous day,week,...	f / b
goto today / some date (prompt)	. / j

Remote editing

digit argument	0-9
change state of current TODO item	t
kill item and source	C-k
archive default	\$ / a
refile the subtree	C-c C-w
set/show tags of current headline	: / T
set effort property (prefix=nth)	e
set / compute priority of current item	, / P
raise/lower priority of current item	S-UP/DOWN [2]
run an attachment command	C-c C-a
schedule/set deadline for this item	C-c C-s/d
change timestamp one day earlier/later	S-LEFT/RIGHT [2]
change timestamp to today	>
insert new entry into diary	i

start/stop/cancel the clock on current item	I / O / X
jump to running clock entry	J
mark / unmark / execute bulk action	m / u / B

Misc

follow one or offer all links in current entry	C-c C-o
--	---------

Calendar commands

find agenda cursor date in calendar	c
compute agenda for calendar cursor date	c
show phases of the moon	M
show sunrise/sunset times	S
show holidays	H
convert date to other calendars	C

Quit and Exit

quit agenda, remove agenda buffer	q
exit agenda, remove all agenda buffers	x

LaTeX and cldatex-mode

preview LaTeX fragment	C-c C-x C-1
expand abbreviation (cldatex-mode)	TAB
insert/modify math symbol (cldatex-mode)	' / '
insert citation using RefTeX	C-c C-x [

Exporting and Publishing

Exporting creates files with extensions *.txt* and *.html* in the current directory. Publishing puts the resulting file into some other place.

export/publish dispatcher	C-c C-e
toggle asynchronous export	C-c C-e C-a
toggle body/visible only export	C-c C-e C-b/v
toggle subtree export	C-c C-e C-s
insert template of export options	C-c C-e #
toggle fixed width for entry or region	C-c :
toggle pretty display of scripts, entities	C-c C-x \

Comments: Text not being exported

Lines starting with # and subtrees starting with COMMENT are never exported.

toggle COMMENT keyword on entry	C-c ;
---------------------------------	-------

Dynamic Blocks

update dynamic block at point	C-c C-x C-u
update all dynamic blocks	C-u C-c C-x C-u

Notes

[1] This is only a suggestion for a binding of this command. Choose your own key as shown under ACTIVATION.

[2] Keybinding affected by `org-support-shift-select` and also `org-replace-disputed-keys`.

Copyright © 2018 Free Software Foundation, Inc.
v9.1.9 for Org-Mode 9.1.9, 2018

Author: Philip Rooke
based on refcard design and format by Stephen Gildea
Released under the terms of the GNU General Public License
version 3 or later.

For more Emacs documentation, and the TeX source for this card, see the Emacs distribution, or <https://www.gnu.org/software/emacs>