Org-Mode Reference Card (1/2)

(for version 4.66)

Getting Started

```
(add-to-list 'auto-mode-alist '("\\.org$" . org-mode))
(define-key global-map "\C-cl" 'org-store-link)¹
(define-key global-map "\C-ca" 'org-agenda)¹
```

For the many customization options try	M-x org-customize
To read the on-line documentation try	M-x org-info

Visibility Cycling

rotate current subtree between states	TAB
rotate entire buffer between states	S-TAB
show the whole file	C-c C-a
reveal context around point	C-c C-r
show subtree in indirect buffer, ded. frame	C-c C-k

Motion

next/previous heading	C-c C-n/p
next/previous heading, same level	C-c C-f/b
backward to higher level heading	C-c C-u
jump to another place in document	C-c C-j
previous/next plain list item	S-UP/DOWN³

Structure Editing

insert new heading/item at current level	M-RET
insert new TODO entry/checkbox item	M-S-RET
promote current heading up one level	M-LEFT
demote current heading down one level	M-RIGHT
promote current subtree up one level	M-S-LEFT
demote current subtree down one level	M-S-RIGHT
move subtree/list item up/down	M-S-UP/DOWN
kill subtree	C-c C-x C-w
copy subtree	C-c C-x M-w
yank subtree	C-c C-x C-y
narrow buffer to current subtree	C-x n s

Archiving

toggle ARCHIVE tag	C-c C-x C-a
mark fully done children	C-u C-c C-x C-a
force cycling of an ARCHIVEd tree	C-TAB
move subtree to archive file	C-c C-x C-s
move all fully done children	C-u C-c C-x C-s

To set archive location for current file, add a line like²:
`#+ARCHIVE: archfile::* Archived Tasks`

Filtering and Sparse Trees

show sparse tree for all matches of a regexp	C-c /
view TODO's in sparse tree	C-c C-v
global TODO list in agenda mode	C-c t¹
create sparse tree with all deadlines due	C-c C-w
time sorted view of current org file	C-c a L

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	C-c C-c
re-align the table, move to next field	TAB
move to previous field	S-TAB
re-align the table, move to next row	RET

Row and column editing

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

Regions

cut rectangular region	C-c C-x C-w
copy rectangular region	C-c C-x M-w
paste rectangular region	C-c C-x C-y
fill paragraph across selected cells	C-c C-q

Calculations (in combination with Emacs `calc` package)

set and eval column formula	C-c =
set and eval field formula	C-u C-c =
edit formulas in separate buffer	C-c '
re-apply all stored equations to current line	C-c *
re-apply all stored equations to entire table	C-u C-c *
iterate table to stability	C-u C-u C-c *
rotate calculation mark through # * ! ^_ $	C-#
show line, column, formula reference	C-c ?
sum numbers in current column/rectangle	C-c +
copy down with increment	S-RET³

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

Example: Add Col1 and Col2	=$1+$2
... with printf format specification	=$1+$2;%.2f
... with constants from constants.el	=$1/$c/$cm
sum from 2nd to 3rd hline	:=vsum(@II..@III)
apply current column formula	=

Miscellaneous

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

Tables created with the `table.el` package

insert a new `table.el` table	C-c ~
recognize existing table.el table	C-c C-c
convert table (Org-mode ↔ table.el)	C-c ~

Links

globally store link to the current location	C-c l¹
insert a link (TAB completes stored links)	C-c C-l
insert file link with file name completion	C-u C-c C-l
edit (also hidden part of) link at point	C-c C-l
open file links in emacs	C-c C-o
...force open in emacs/other window	C-u C-c C-o
open link at point	mouse-1/2
...force open in emacs/other window	mouse-3
record a position in mark ring	C-c %
jump back to last followed link(s)	C-c &
Find next link	C-c C-x C-n
Find previous link	C-c C-x C-p

Internal Links

<<My Target>>	target
<<<My Target>>>	radio target²
[[*this text]]	find headline
[[this text]]	find target or text in buffer
[[this text][description]]	optional link text

External Links

file:/home/dominik/img/mars.jpg	file, absolute
file:papers/last.pdf	file, relative
file:projects.org::*that text	find headline
file:projects.org::find me	find trgt/string
http://www.astro.uva.nl/~dominik	on the web
mailto:adent@galaxy.net	Email address
news:comp.emacs	Usenet group
bbdb:Richard Stallman	BBDB person
gnus:group	GNUS group
gnus:group#id	GNUS message
vm|wl|mhe|rmail:folder	Mail folder
vm|wl|mhe|rmail:folder#id	Mail message
info:emacs:Regexps	Info file:node
shell:ls *.org	shell command
elisp:(calendar)	elisp form
[[external link][description]]	optional link text

Remember-mode Integration

See the manual for how to make remember.el use Org-mode links and files. The note-finishing command C-c C-c will first prompt for an org file. In the file, find a location with:

rotate subtree visibility	TAB
next heading	DOWN
previous heading	UP

Insert the note with one of the following:

as sublevel of heading at cursor	RET
right here (cursor not on heading)	RET
before current heading	LEFT
after current heading	RIGHT
shortcut to end of buffer (cursor at buf-start)	RET
Abort	q

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 4.66)

TODO Items and Checkboxes

rotate the state of the current item	C-c C-t
view TODO items in a sparse tree	C-c C-v
view 3rd TODO keyword's sparse tree	C-3 C-c C-v
set the priority of the current item	C-c , [ABC]
remove priority cookie from current item	C-c , SPC
raise priority of current item	S-UP[3]
lower priority of current item	S-DOWN[3]
#+SEQ_TODO: TODO TRY BLUFF DONE	todo workflow
#+TYP_TODO: Phil home work DONE	todo types
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
checkbox statistics cookies: insert [/] or [%]	
update checkbox statistics (C-u : whole file)	C-c #

Tags

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

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 C-w
the time between 2 dates in a time range	C-c C-y
change timestamp at cursor by ±1 day	S-RIGHT/LEFT[3]
change year/month/day at cursor by ±1	S-UP/DOWN[3]
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 clock on current item	C-c C-x C-o
cancel current clock	C-c C-x C-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

LaTeX and cdlatex-mode

preview LaTeX fragment	C-c C-x C-l
Expand abbreviation (cdlatex-mode)	TAB
Insert/modify math symbol (cdlatex-mode)	' / '

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-'
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 in agenda files	C-c a m[1]
match tags in TODO entries	C-c a M[1]
find stuck projects	C-c a #[1]
show timeline of current org file	C-c a L[1]
configure custom commands	C-c a C[1]
configure stuck projects	C-c a ![1]
agenda for date at cursor	C-c C-o

To set categories, add lines like[2]:
#+CATEGORY: MyCateg

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	b
toggle follow-mode	f

Change display

delete other windows	o
switch to daily / weekly view	d / w
toggle inclusion of diary entries	D
toggle time grid for daily schedule	g
toggle display of logbook entries	l
refresh agenda buffer with any changes	r
save all org-mode buffers	s
display the following org-agenda-ndays	RIGHT
display the previous org-agenda-ndays	LEFT
goto today	.

Remote editing

digit argument	0-9
change state of current TODO item	t
kill item and source	C-k
archive the subtree	$
show tags of current headline	T
set tags for current headline	:
toggle ARCHIVE tag	a
set priority of current item	p
raise/lower priority of current item	S-UP/DOWN[3]
display weighted priority of current item	P
schedule/set deadline for this item	C-c C-s/d
change timestamp one day earlier/later	S-LEFT/RIGHT[3]
change timestamp to today	>
insert new entry into diary	i
start the clock on current item (clock-in)	I
stop the clock (clock-out)	O
cancel current clock	X

Misc

Open link in current line	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

Calendar and Diary Integration

Include Emacs diary entries into Org-mode agenda with:

(setq org-agenda-include-diary t)

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
export visible part only	C-c C-e v
insert template of export options	C-c C-x t
toggle fixed width for entry or region	C-c :

HTML formatting

make words **bold**	*bold*
make words *italic*	/italic/
make words underlined	_underlined_
sub- and superscripts	x^3, J_dust
TEX-like macros	\alpha, \to
typeset lines in fixed width font	start with :
tables are exported as HTML tables	start with	
links become HTML links	http:... etc
include html tags	@...@

Comments: Text not being exported

Text before the first headline is not considered part of the document and is therefore never exported. Lines starting with # are comments and are not exported. Subtrees whose header starts 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 you own key as shown under INSTALLATION.

[2] After changing a #+KEYWORD or <<<target>>> line, press C-c C-c with the cursor still in the line to update.

[3] Keybinding affected by org-CUA-compatibility.