Org-Mode Reference Card (1/2)

(for version 3.18)

Getting Started

Put the following in your `~/.emacs`[1]

```
(autoload 'org-mode "org" "Org mode" t)
(autoload 'org-diary "org" "Org mode diary entries")
(autoload 'org-agenda "org" "Agenda from Org files" t)
(autoload 'org-store-link "org" "Store org link" t)
(autoload 'orgtbl-mode "org" "Orgtbl minor mode" t)
(autoload 'turn-on-orgtbl "org" "Orgtbl minor mode")
(add-to-list 'auto-mode-alist '("\\.org$" . org-mode))
(define-key global-map "\C-cl" 'org-store-link)[2]
(define-key global-map "\C-ca" 'org-agenda)[2]
```

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`

Motion

next heading	`C-c C-n`
previous heading	`C-c C-p`
next heading, same level	`C-c C-f`
previous heading, same level	`C-c C-b`
backward to higher level heading	`C-c C-u`
jump to another place in document	`C-c C-j`

Structure Editing

insert new heading at same level as current	`M-RET`
insert new TODO entry	`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 up	`M-S-UP`
move subtree down	`M-S-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`
archive subtree	`C-c $`

To set archive location for current file, add a line like[3]:

```
#+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`
create sparse tree with all deadlines due	`C-c C-w`
time sorted view of current org file	`C-c C-r`
agenda for the week	`C-c a1`
agenda for date at cursor	`C-c C-o`

TODO Items

rotate the state of the current item	`C-c C-t`
view TODO items in a sparse tree	`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`[4]
lower priority of current item	`S-DOWN`[4]

per-file TODO workflow states: add line(s) like[3]:

```
#+SEQ_TODO: TODO PROCRASTINATE BLUFF DONE
```

per-file TODO keywords: add line(s) like[3]:

```
#+TYP_TODO: Phil home work DONE
```

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-LEFT`[4]
change timestamp at cursor by $+1$ day	`S-RIGHT`[4]
change year/month/day at cursor by -1	`S-DOWN`[4]
change year/month/day at cursor by $+1$	`S-UP`[4]
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`
While prompted for a date:	
... select date in calendar	`mouse-1/RET`
... scroll calendar back/forward one month	`< / >`
... forward/backward one day	`S-LEFT/RIGHT`
... forward/backward one week	`S-UP/DOWN`
... forward/backward one month	`M-S-LEFT/RIGT`

Links

globally store link to the current location	`C-c l`[2]
insert a link (TAB completes stored links)	`C-c C-l`
insert file link with file name completion	`C-u C-c C-l`
open link at point	`C-c C-o`
open file links in emacs	`C-u C-c C-o`
open link at point	`mouse-2`
open file links in emacs	`mouse-3`

Link types

`<http://www.astro.uva.nl/~dominik>`	on the web
`<mailto:adent@galaxy.net>`	EMail address
`<news:comp.emacs>`	Usenet group
`<file:/home/dominik/img/mars.jpg>`	file, absolute
`<file:papers/last.pdf>`	file, relative
`<file:~/code/main.c:255>`	file with line nr.
`<bbdb:Richard Stallman>`	BBDB person
`<shell:ls *.org>`	shell command
`<gnus:group>`	GNUS group
`<gnus:group#id>`	GNUS message
`<vm:folder>`	VM folder
`<vm:folder#id>`	VM message
`<vm://myself@some.where.org/folder#id>`	VM remote

Wanderlust `<wl:...>` and RMAIL `<rmail:...>` like VM

Tables

Creating a table

insert a new Org-mode table	`M-x org-table-create`
... or just start typing, e.g.	`|Name|Phone|Age RET |- TAB`
convert region to table	`C-c C-c`
... separator at least 3 spaces	`C-3 C-c 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 horizontal line below the current row	`C-c -`
insert horizontal line above the current row	`C-u 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

Except for the summation commands, these need the Emacs calc package installed.

set and eval column formula	`C-c =`
set and eval named-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 *`

`TAB`, `RET` and `C-c C-c` trigger automatic recalculation in lines starting with: `| # |`.

rotate calculation mark through # * ! ^ _ $	`C-#`
display column number cursor is in	`C-c ?`
sum numbers in current column/rectangle	`C-c +`
copy down with increment	`S-RET`[4]

A formula can also be typed directly into into a field and will executed by `TAB`, `RET` and `C-c C-c`. A leading `=` 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 3rd hline above to here	`:=vsum(&III)`
apply current column formula	`=`

Miscellaneous

toggle visibility of vertical lines	`C-c |`
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 ~`

Org-Mode Reference Card (2/2)

(for version 3.18)

Timeline and Agenda

show timeline of current org file	`C-c C-r`
... include past dates	`C-u C-c C-r`
add current file to your agenda	`C-c [`
remove current file from your agenda	`C-c ]`
compile agenda for the current week	`C-c a`[2]
agenda for date at cursor	`C-c C-o`

To set category for current file, add line[3]:
`#+CATEGORY: MyCateg`

Commands available in an agenda buffer

The agenda buffer is electric: single key presses execute commands.

View org file

show original location of item	`SPC`
... also available with	`mouse-3`
show and recenter window	`l`
goto original location in other window	`TAB`
... also available with	`mouse-2`
goto original location, delete other windows	`RET`
toggle follow-mode	`f`

Change display

delete other windows	`o`
switch to weekly view	`w`
switch to daily view	`d`
toggle inclusion of diary entries	`D`
toggle time grid for daily schedule	`g`
refresh agenda buffer with any changes	`r`
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`
set priority of current item	`p`
raise priority of current item	`S-UP`[4]
lower priority of current item	`S-DOWN`[4]
display weighted priority of current item	`P`
change timestamp to one day earlier	`S-LEFT`[4]
change timestamp to one day later	`S-RIGHT`[4]
change timestamp to today	`>`
insert new entry into diary	`i`

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`

Exporting

Exporting creates files with extensions *.txt* and *.html* in the current directory.

export as ASCII file	`C-c C-x a`
export visible text only (e.g. for printing)	`C-c C-x v`
export as HTML file	`C-c C-x h`
export as HTML and open in browser	`C-c C-x b`
prefix arg sets nb. of headline levels, e.g.	`C-3 C-c C-x h`
export as iCalendar file	`C-c C-x i`
export all agenda files as iCalendar files	`C-c C-x C-i`
compine all agenda files to single iCal file	`C-c C-x C-c`
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	`@<b>...@</b>`

Export options

Include additional information for export by putting these anywhere in the org file. Use `M-TAB` completion to make sure to get the right keywords. `M-TAB` again just after keyword is complete inserts examples.

the title to be shown	`#+TITLE:`
the author	`#+AUTHOR:`
authors email address	`#+EMAIL:`
language code for html	`#+LANGUAGE:`
free text description of file	`#+TEXT:`
... which can carry over multiple lines	`#+TEXT:`
settings for the export process - see below	`#+OPTIONS:`

Settings for the OPTIONS line

set number of headline levels for export	`H:2`
turn on/off section numbers	`num:t`
turn on/off table of contents	`toc:t`
turn on/off linebreak preservation	`\n:nil`
turn on/off quoted html tags	`@:t`
turn on/off fixed width sections	`::t`
turn on/off tables	`|:t`
turn on/off TEX syntax for sub/super-scripts	`^:t`
turn on/off emphasised text	`*:nil`
turn on/off TEX macros	`TeX:t`

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 ;`

Completion

In-buffer completion completes TODO keywords at headline start, TeX macros after "\", option keywords after "`#-`", and dictionary words elsewhere.

Complete word at point	`M-TAB`

Calendar and Diary Integration

To include entries from the Emacs diary in your Org-mode agenda:

```
(setq org-agenda-include-diary t)
```

To include your Org-mode agenda in your normal diary, make sure you're using the fancy diary display and in your `~/diary` file add

```
&%%(org-diary)
```

to include all the files listed in `org-agenda-files`. For more selective file inclusion use a line for each file:

```
&%%(org-diary) ~/path/to/some/org-file.org
```

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`

CUA and pc-select compatibility

Configure the variable `org-CUA-compatibility` to make Org-mode avoid the `S-<cursor>` bindings used by these modes. When set, Org-mode will change the following keybindings (also in the agenda buffer, but not during date selection). See note mark four[4] throughout the reference card.

```
S-UP    → M-p          S-DOWN  → M-n
S-LEFT  → M--          S-RIGHT → M-+
S-RET   → C-S-RET
```

Notes

[1] The six autoload forms are only needed if Org-mode is not part of Emacs, or an XEmacs package.

[2] This is only a suggestion for a binding of this command. Choose you own key as shown under INSTALLATION.

[3] After changing a `#+KEYWORD` line, press `C-c C-c` with the cursor still in a line to make Org-mode notice the change.

[4] Keybinding affected by `org-CUA-compatibility`.

Copyright © 2005 Free Software Foundation, Inc.
v3.18 for Org-Mode 3.18, 2005
Author: Philip Rooke
based on refcard design and format by Stephen Gildea

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies.