

# Org-Mode Beginners Customization Guide

Carsten Dominik

April 11, 2017

## Contents

## Introduction

Org-mode is a highly customizable package. It currently contains close to 400 customization variables that can be changed to tweak every detail, and more than 260 are known to be actually used out there.

However, as a beginner you do not care about this kind of flexibility. On this page, we have a list of five settings that you might want to try first in order to personalize your system.

Once you are done with that, we also have a list of some 40 variables that are changed by many users.

## The Basics

### Minimal customization

The minimal customization needed to use Org-mode is – **Nothing at all!**

Org-mode works out of the box, and besides the steps described in the manual to activate it, *nothing is needed at all*. Just open a *.org* file, press **C-c** **[** to tell org that this is a file you want to use in your agenda, and start putting your life into plain text.

OK, for completeness, let's just repeat what is needed to activate Org-mode in files with *.org* extension, and a few important key assignments.

```
(add-to-list 'auto-mode-alist '("\\.org\\'" . org-mode))
(global-set-key "\C-cl" 'org-store-link)
(global-set-key "\C-ca" 'org-agenda)
;; (global-font-lock-mode 1) Not needed in recent emacsen
```

## Five small steps toward a personalized system

### One: More TODO keywords

Define the TODO states you find useful and single letters for fast selection. Customize the variable `org-todo-keywords` or simply do this right in the file with<sup>1</sup>:

```
#+TODO: TODO(t) STARTED(s) WAITING(w) | DONE(d) CANCELED(c)
```

### Two: Which tags do you use most?

You can always add tags freely and by hand, but if you configure the most important ones along with fast-access keys, life will be better. Configure the variable `org-tags-alist` or simply do this right in the file with<sup>1</sup>

```
#+TAGS: home(h) work(w) @computer(c) @phone(p) errands(e)
```

### Three: Which files are relevant for the agenda?

When Org compiles agenda views like the the agenda for the current week (`C-c a a`) or the global TODO list (`C-c a t`), it checks all files in the variable `org-agenda-files`. Instead of setting this variable explicitly, it is much easier to just add and remove the current buffer with `C-c [` and `C-c ]`, respectively.

### Four: Find what you need to see now

If you need to sharpen the predefined agenda commands, define your own Agenda commands, using the variable `org-agenda-custom-commands`. This is a pretty complex variable, but if you use the customize interface<sup>2</sup>, it is not too hard. Everyone ends up customizing this one after getting comfortable with Org-mode. Do checkout this tutorial on building your own custom agenda commands as well.

### Five: Capture ideas with predefined templates

Use `org-capture` to quickly capture ideas, tasks, and notes. Populate<sup>3</sup> the variable `org-capture-templates` with templates and target locations.

---

<sup>1</sup>press `C-c C-c` in the line after changing it

<sup>2</sup>`M-x customize-variable RET org-agenda-custom-commands RET`

<sup>3</sup>`M-x customize-variable RET org-capture-templates RET`

## Pretend to be a power-user

Here is a list of the variables most frequently changed by power users. The variables mentioned above are repeated in this list. Yes, some power users change 60 options or more, but these showed up most frequently in the customization survey.

### Basic setup

Here are a few things about basic setup that many people change.

**org-directory** Where are my Org files typically located? Org actually uses this variable only under rare circumstances, like when filing remember notes in an interactive way and prompting you for an Org file to put the note into.

**org-agenda-files** Which files do I want to be checked for entries when compiling my agenda? Many people do not customize this one, but just use `C-c [` and `C-c ]` to add or remove the current file, respectively.

**org-startup-folded** When visiting an Org file, in what folding state do I first want to see it? Many use `#+STARTUP` options to set this on a per-file basis.

**org-archive-location** When archiving an entry, where will it go?

### Editing behavior and appearance

Besides being an organizer, Org-mode is also a text mode for writing and taking notes. The following variables that influence basic editing behavior and the appearance of the buffer are often customized:

**org-hide-leading-stars** Make the outline more list-like by hiding all leading stars but one.

**org-odd-levels-only** Should 2 stars be added per level? This makes the indentation more like in a book.

**org-special-ctrl-a/e** Should `C-a` and `C-e` behave specially, considering the headline and not the leading stars, todo keywords, or the trailing tags? About equal numbers of users set this to `t` or to `reversed`

**org-special-ctrl-k** Should `C-k` behave specially in headlines, considering tags and visibility state?

**org-completion-use-ido** Should *ido.el* be used for completion whenever it makes sense?

**org-return-follows-link** Should pressing RET on a hyperlink follow the link? People who are used to this from web browsers often make this choice.

**org-blank-before-new-entry** Org-mode tries to be smart about inserting blank lines before new entries/items, by looking at what is before the previous entry/item. Customize this to out-smart it.

### The TODO keywords

**org-todo-keywords** Which TODO keywords should be used? Also you can define keys for fast access here. Very many people use this, or define the keywords with a `#+TODO:` setting in the buffer.

**org-todo-keyword-faces** Here you can define different faces for different TODO keywords.

**org-enforce-todo-dependencies** Should unfinished children block state changes in the parent?

**org-enforce-todo-checkbox-dependencies** Should unfinished checkboxes block state changes in the parent?

### Tags

**org-tag-alist** Which tags should be available? Note that tags besides the configured ones can be used, but for the important ones you can define keys for fast access here.

**org-tags-column** How should tags be aligned in the headline?

**org-fast-tag-selection-single-key** Set this to make the tags interface even faster, if all you normally do is changing a single tag.

### Progress logging

**org-log-done** Do you want to capture time stamps and/or notes when TODO state changes, in particular when a task is DONE? A simple setting that many use is `(setq org-log-done 'time)`.

## Capture and Refile

**org-reverse-note-order** When adding new entries (or tasks) to a list, do I want the entry to be first or last in the list?

**org-capture** is great for fast capture of ideas, notes, and tasks. It is one of the primary capture methods in Org-mode.

**org-capture-templates** Prepare templates for the typical notes and tasks you want to capture quickly. I believe everyone using **org-capture** customizes this.

**org-default-notes-file** If you do not set up templates with target files, at least tell Org where to put captured notes.

*Refiling* means moving entries around, for example from a capturing location to the correct project.

**org-refile-targets** What should be on the menu when you refile tasks with `C-c C-w`?

**org-refile-use-outline-path** How would you like to select refile targets. Headline only, or the path along the outline hierarchy?

## Agenda Views

**org-agenda-start-on-weekday** Should the agenda start on Monday, or better today?

**org-agenda-ndays** How many days should the default agenda show? Default is 7, a whole week.

**org-agenda-include-diary** Should the agenda also show entries from the Emacs diary?

**org-agenda-custom-commands** Define your own Agenda commands. Complex, advanced variable, but pretty much everyone ends up configuring it. Use `customize` to configure it, this is the best and safest way. Do checkout this tutorial on building your own custom agenda commands as well.

**org-agenda-sorting-strategy** How should things be sorted in the agenda display. Even though I think the defaults are very usable, power users tend to tweak this.

**org-stuck-projects** How to find projects that need attention?

To reduce clutter in the task list for today, many users like to remove tasks from the daily list right when they are done. The following variables give detailed control to what kind of entries this should apply:

**org-agenda-skip-scheduled-if-done** Scheduled entries. Many users turn this on.

**org-agenda-skip-deadline-if-done** Deadlines. Many users turn this on.

**org-agenda-skip-timestamp-if-done** Entries with any timestamp, appointments just like scheduled and deadline entries. Relatively few users select this one.

People who use Org like a day planner, who schedule all tasks to specific dates, often like to not have scheduled tasks listed in their global TODO list, because scheduling it already means to have taking care of it in a sense, and because they know they will run into these tasks in the agenda anyway.

**org-agenda-todo-ignore-deadlines** Don't show deadline tasks in global TODO list.

**org-agenda-todo-ignore-with-date** Don't show any tasks with a date in the global TODO list.

**org-agenda-todo-ignore-scheduled** Don't show scheduled tasks in the global TODO list.

### **Export/Publishing setup**

**org-export-with-L<sup>A</sup>T<sub>E</sub>X-fragments** Should L<sup>A</sup>T<sub>E</sub>X fragments be converted to inline images for HTML output?

**org-export-html-style** Customize the default style for HTML export.

**org-publish-project-alist** Set up projects that allow many files to be exported and published with a single command.

## Become a true power user

If you want to become a true power user, see for yourself what some users do. The Emacs customization system<sup>4</sup> organizes all variables into a structure that can be used to easily find the one particular option you might be looking for. Also, the Org-mode manual and the FAQ mention many variables in the appropriate context.

---

<sup>4</sup>M-x `org-customize` RET