

org 实践

aborn

2016-10-12

Contents

1	org 模式简介	1
2	文档结构	1
2.1	目录结构	1
2.2	显示与隐藏	1
2.3	列表	2
2.4	块结构	2
3	表格	2
4	超链接	2
4.1	链接格式	2
4.1.1	内部链接	2
4.1.2	外部链接	3
4.1.3	链接处理相关命令	3
5	待办事项	3
6	日程表 (Agenda View)	3
6.1	日程文件 (Agenda files)	3
6.2	分发按钮	3
6.3	内建 Agenda 视图	4
6.4	计划 Schedule	4
7	Org 快速记录	4
7.1	如何使用 org-capture?	4
7.2	org 条目复制与移动	4
7.3	记录模板	4

7.3.1	快捷键	5
7.3.2	描述	5
7.3.3	类型	5
7.3.4	目标文件	5
7.3.5	模板	5
7.3.6	属性 properties	6
8	Org 的导出功能	6
8.1	导出的 Dispatcher	6
9	org-capture.el	6

1 org 模式简介

Emacs 的 org-mode 可用于记笔记、管理自己的待办事项 (TODO lists)，同时，也可用于管理项目。它是一个高效的纯文本编辑系统。

2 文档结构

Org 是基于 Outline-mode，并提供灵活的命令编辑结构化的文档。其文档结构语法跟 markdown 很类似。

2.1 目录结构

Org 的目录结构在每行最左边以星号标记，星号越多，标题层级越深。下面是一些例子：

```
\* 一级目录
\** 二级目录
\*** 三级目录

\* 另一个一级目录
```

2.2 显示与隐藏

目录结构下的内容可以隐藏起来，通常用采用 **TAB** 和 *S-TAB* 这两个命令来切换。

2.3 列表

Org 提供三种类型的列表：有序列表、无序列表和描述列表

1. 有序列表以'1.' 或者'1)'
2. 无序列表以'-'、'+' 或者'*'
3. 描述列表

2.4 块结构

在 Org 文档中，加入代码块这种类型的块结构，都是采用 begin...end 这种模式，下面是一个例子：

```
\#+BEGIN_EXAMPLE  
\#+END_EXAMPLE
```

3 表格

4 超链接

Org 模式提供了比较好用的超链接方式，可以链接到普通网页、文件、email 等。

4.1 链接格式

Org 模式支持两种链接，即，内部链接和外部链接。它们有相同的格式：

[[链接]][描述]] 或 当只有链接没有描述 [[链接]]

一旦链接编辑完成，在 org 模式下，只显示 **描述**部分，而不会显示整体（后一种是只显示链接）。为了编辑链接和描述，需要通过快捷键 **C-c C-l** 来完成（注意：编辑结束后按 Enter 完成修改操作）。

4.1.1 内部链接

内部链接是指向当前文件的链接，它的链接格式：

[[#链接ID]]

其中 **链接 ID** 是文档中唯一的标识 ID

4.1.2 外部链接

Org 支持的外部链接有很多中形式，如文件、网页、新闻组、电子邮件信息、BBDB 数据条目等。它们以一个短的标识字符串打头，紧接着是一个冒号，冒号后面没有空格字符。

4.1.3 链接处理相关命令

- `org-store-link` 保存当前位置的一个链接，以备后面插入使用
- `org-insert-link` 插入链接，绑定的快捷键为 `C-c C-l`，如果光标正在一个链接上，那么 `C-c C-l` 的行为是编辑这个链接及其描述

5 待办事项

Org 模式用来管理自己的 TODO list 非常方便

6 日程表 (Agenda View)

我们可以用 Org 来安排自己的行程

6.1 日程文件 (Agenda files)

变量 `org-agenda-files` 保存了一个文件列表，这些文件用来记录日程，下面是一些操作函数：`C-c [` 将当前文件加入到 agenda 文件列表最前页面 `org-agenda-file-to-front C-c]` 将当前文件从 agenda 文件列表中删除 `org-remove-file`

6.2 分发按键

默认采用 `C-c a`，接下的默认的命令有：

- `a` 创建一个日程
- `t/T` 创建一个 TODO items
- `L` 对当前文件生成 timeline

6.3 内建 Agenda 视图

6.4 计划 Schedule

用 org 来安排日程

- org-schedule 将当前 TODO 添加计划时间

7 Org 快速记录

有时候，突然想到一些待办事项，或者一些突发的灵感。这时，我们想用 emacs 快速记录它，Org-Capture 提供这个好用的功能。它的前身是 org-remember.el（注：从 org 8.0 开始，org-remember 被 org-capture）替代。

7.1 如何使用 org-capture?

快速记录的命令为 **M-x org-capture**，默认绑定的快捷键为 **C-c c**。当这个命令被调用后，你可以使用自己定义好的模板快速创建记录。一旦完成内容的输入，按下 **C-c C-c** (org-capture-finalize)，来完成。然后，你就能继续做你当下的事。如果想跳转到刚刚创建的记录的 buffer，用 **C-u C-c C-c** 来完成。如果想中途中止输入，只要按下 **C-c C-k** (org-capture-kill)。

7.2 org 条目复制与移动

有时候，我们想将当前的某条目转移到其他文件或者其他项目里。这时，我们会用到 org-copy 和 org-refile 这两个命令。它们对应的快捷键分别是 **C-c M-w** 及 **C-c C-w**。这里有一个问题是，目标文件如何配置？目录文件的配置由一个变量决定，**org-refile-targets**，我自己的配置如下：

```
(setq org-refile-targets
      '((nil :maxlevel . 3)          ;; 当前文件的最大层级
        (aborn-gtd-files :maxlevel . 3)))
```

注意：我这时将文件放在 aborn-gtd-files 文件列表里。

7.3 记录模板

记录的模板为一个列表变量，org-capture-templates，列表的每条记录由如下几段组成：

```
("t" "Todo" entry (file+headline (expand-file-name org-default-notes-file org-directory)
                                   "* TODO %?\n  创建于:%T  %i\n"))
```

7.3.1 快捷键

如例子中的那样，”t”表示对应按键 t 这个快捷键。它能帮助我们快速地选中哪条模板进行快速记录。

7.3.2 描述

接下来是一段简单的描述

7.3.3 类型

第三段表示类型，有五种类型：entry item checkitem table-line plain

- entry 普通的 Org 结点，保证目标文件为 org-mode 文件，插入的时候将作为目录结点的子结点

(如果没有，将做为顶级结点)；

- item 与 entry 类似，不同点在于它的目标文件可以为简单的纯文本文件；
- checkitem 复选条目；
- table-line 在目标文件中的第一个 table 中插入新行；
- plain 纯文本记录

7.3.4 目标文件

第四个字段配置目标文件

7.3.5 模板

第五个字段表示模板，模板参数 含义如下：

- %t 只有日期的时间戳
- %T 日期 + 时间的时间戳
- %u,%U 如上，只不过它们是 inactive 的
- %i 初始化文本，当前上下文将作为初始化文本

7.3.6 属性 properties

最后一个字段表示属性列表，支持以下属性配置：

- `:prepend` 一般一个记录条目插入在目标文件的最后，这个属性可以将条目插入在最前
- `:immediate-finish` 立刻完成，没有交互
- `:clock-in` 对这个条目设置闹钟
- `:kill-buffer` 如果目标文件没有相应的访问 buffer，插入后，自动关闭 buffer

8 Org 的导出功能

Org 文件支持导出多种格式的目标文件，如 ASCII 文件、HTML 文件 (用于发布为 Web)、PDF 文档等。

8.1 导出的 Dispatcher

任何导出命令都有一个前缀按键，我们称之为 Dispatcher，为 **C-c C-e**

9 org-capture.el

Org 8.0 以后版本采用 `org-capture.el` 取代原有的 `org-remember.el`