

# EMACS NOTES

hexinzheng

2020 年 2 月 13 日

## 目录

<b>1 问题</b>	<b>3</b>
<b>2 调试脚本</b>	<b>4</b>
<b>3 澄清概念</b>	<b>4</b>
<b>4 重要概念</b>	<b>4</b>
4.1 屏幕 (Screen)	4
4.1.1 Point	4
4.1.2 Echo Area	4
4.1.3 Mode Line	4
4.2 用户输入 (User input)	5
4.3 输入键 (Keys)	5
4.4 命令 (Command)	5
4.5 进入 Emacs	5
4.6 退出 Emacs	5
<b>5 基本编辑命令</b>	<b>6</b>
5.1 基础	6
5.1.1 插入文本 (Insert Text)	6
5.1.2 移动光标 (Move Point)	6
5.1.3 删除 (Erasing)	6
5.1.4 基本撤销 (Basic Undo)	6
5.1.5 文件 (Files)	6
5.1.6 帮助 (Help)	6
5.1.7 空行 (Blank Lines)	7
5.1.8 连续行 (Continuation Lines)	7
5.1.9 位置信息 (Position Info)	7
5.1.10 参数 (Arguments)	7
5.1.11 重复 (Repeating)	7

5.2	Minibuffer . . . . .	7
5.3	M-x . . . . .	7
5.4	帮助 (help) . . . . .	7
<b>6</b>	<b>org-mode</b> . . . . .	<b>7</b>
6.1	Agenda Views . . . . .	7
6.1.1	Agenda Files . . . . .	7
6.2	Document Structure . . . . .	8
6.2.1	Headlines . . . . .	8
6.3	ToDo Items . . . . .	8
6.4	Datetimes . . . . .	8
6.4.1	Timestamps . . . . .	8
6.4.2	创建时间戳 . . . . .	8
6.4.3	Deadlines 和 Scheduling . . . . .	9
<b>7</b>	<b>Timer Example</b> . . . . .	<b>10</b>
7.1	Clock column view example . . . . .	10
7.2	Effort estimate example . . . . .	10
<b>8</b>	<b>Orgmode (Studing)</b> . . . . .	<b>10</b>
8.1	E01S01 : Headlines & outline mode . . . . .	10
8.2	E01S03 : Schedule, Deadlines & Agenda views . . . . .	10
8.3	E01S04 : Repeating tasks . . . . .	10
8.3.1	规律重复出现的时间戳 . . . . .	10
8.4	E01S05 : Checklists [2/4] . . . . .	10
8.5	E02S01 : Tags . . . . .	11
8.6	E02S02 : Agenda view (advanced) . . . . .	11
8.7	E02S03 : Customized Agenda view . . . . .	11
8.8	E02S04 : Drawers,Logging & quick notes . . . . .	11
8.9	E02S05 : Archiving . . . . .	12
8.10	E03S01 : Automatic logging of status changes . . . . .	12
8.11	E03S02 : Splitting your system up to several files . . . . .	12
8.12	E03S03 : The first capture template(s) . . . . .	12
8.13	E03S04 : The -PROPERTIES - drawer . . . . .	13
8.14	E03S05 : Archiving to different files . . . . .	13
8.15	E04S01 : Ordered tasks . . . . .	13
8.16	E04S02 : Timers . . . . .	13
8.17	E04S03 : Clocking (aka time tracking) . . . . .	13
8.17.1	Football notes . . . . .	13
8.17.2	Clock Tracking example . . . . .	13
8.18	E04S04 : Column view . . . . .	14

8.19	E04S05 : Effort estimates . . . . .	14
8.20	E05S01 : Linking (internal) . . . . .	14
8.21	E05S02 : Linking (external) . . . . .	15
8.22	E05S03 : Attachments . . . . .	15
8.23	E05S04 : Priorities . . . . .	15
8.24	E05S05 : Tables . . . . .	15
8.25	E06S01 : Exporting . . . . .	15
8.25.1	export chinese pdf ? . . . . .	15
8.26	E06S02 : Advanced exporting . . . . .	15
8.27	E06S03 : Publishing . . . . .	16
8.28	E06S04 : Dynamic blocks . . . . .	16
8.29	E06S05 : Tracking habits . . . . .	16
8.30	E07S01 : Bulk agenda actions . . . . .	16
8.31	E07S02 : Presenting my system . . . . .	16
8.32	E07S03 : Google Calendar integration . . . . .	16
8.33	E07S04 : Source code in OrgMode . . . . .	16
8.34	E07S05 : . . . . .	16
<b>9</b>	<b>Use Emacs</b>	<b>16</b>
9.1	62 : Magit . . . . .	16
<b>10</b>	<b>Git</b>	<b>16</b>
10.1	Git Basic . . . . .	16
10.1.1	in master . . . . .	16
10.1.2	switch to branch . . . . .	16
10.1.3	switch back to master and merge . . . . .	17
10.1.4	git to remote . . . . .	17
10.1.5	get newest repo . . . . .	17
10.2	magit . . . . .	17

## 1 问题

- ☐ org 转换为标准 latex 文件
- ☐ org 输出为 html 文件
- ☐ 使用 github 样式显示文件
- ☒ 在 github 上建立自己的站点
- ☒ 默认浏览器改为 qutebrowser
- ☐ 安装 emacs 26.1 。主要是 emacs-26-non-common-dfsg.

☐ Open org-mode html in EWW.

## 2 调试脚本

- 单独加载另外一个 emacs 的初始化文件

```
emacs -q -l ~/youemacs.el
```

```
emacs --no-initial-file --load-file=~/youemacs.el
```

- 调试 elisp 语言,,', 或是 M-x ielm。

## 3 澄清概念

1. 组合键的术语是 Command, 而不是 ShortCut 例如, 搜索插入文件变量的组合键, 关键词应为 command file variable emacs。如使用 shortcut ... 则无法找到有用结果。

## 4 重要概念

### 4.1 屏幕 (Screen)

Emacs 的显示区域称为 Frame, 在 Frame 中可包含多个 Windows。Emacs 中 Frame 在 IDE 中称为 Windows, 而 Emacs 的 Windows 在 IDE 中称为 View。

#### 4.1.1 Point

称为输入提示符号。通过 Cursor 可以改变输入符号的显示。

#### 4.1.2 Echo Area

显示输入命令的区域。Display Custom 修改 Echo Area。Echo Area 用于显示 Minibuffer。退出 Minibuffer 命令是 C-g。

#### 4.1.3 Mode Line

窗口底部是 Mode Line, 显示当前 buffer 状态。Mode Line 文本格式如下

```
cs  ch-fr  buf  pos  line  (major  minor)
```

以下是详细解释

cs	Coding System 的缩写。C-h C unix 给出 unix coding 的具体信息。C-h C utf-8 给出 utf-8 coding 的
ch	表示文件是否保存。* 表示文件未保存，- 表示文件已保存，% 表示为只读文件。
fr	Frame 缩写。F1 为第 1 个 Frame，F2 为第 2 个 Frame。
buf	Buffer name，即当前 Buffer 中文件名。
pos	当前 Buffer 中显示的文件位置。Top 靠近文件首部，Bot 靠近文件尾部，All 显示了全部文件，
line	18:10 表示第 18 行第 10 个字符位置。
major	主编辑模式，如 Text mode，Lisp mode，Latex mode 等。
minor	次编辑模式，可附加到主编辑模式之后。
recursive edit	[...] 表示处于循环编辑模式。

## 4.2 用户输入 (User input)

Emacs 主要设计目的是通过键盘与用户交互，当然 Emacs 也使用鼠标，但这不是设计的出发点。因而要能熟练使用键盘快捷键操作和编辑文件。

## 4.3 输入键 (Keys)

Key 和其组合键会引发 key event。如果一组 Key 引发一条命令，称为 Complete Key。如果无法触发命令，称为 Prefix key，如 C-x 和 M-x。

## 4.4 命令 (Command)

每条命令是一个 Lisp 函数。将命令与组合键绑定在一起称为 Keymaps。C-n 之所以能跳到下一行，是因为绑定了函数 next-line。

## 4.5 进入 Emacs

如果 inhibit-startup-screen 为 non-nil 将不会显示欢迎界面，而直接进入 **scratch** 文件，在其中能运行一些待测试的 Lisp 程序。

如果希望启动 Emacs 时，进入到某个目录或是打开特定文件，可配置 initial-buffer-choice。

## 4.6 退出 Emacs

C-x C-x 退出 Emacs (save-buffers-kill-terminal) C-z Emacs 最小化 (suspend-frame) M-x kill-emacs 退出 Emacs，不需要任何提示

Emacs 能在退出时保存当前会话 Session，下次启动后可先加载此会话。

## 5 基本编辑命令

### 5.1 基础

#### 5.1.1 插入文本 (Insert Text)

C-j [O] 插入新的空行，新行没有 auto-indent。在 Minor Mode 中，可以改变插入方式。例如，Auto Fill Mode 可自动截取超出长度的文本 (参见 Filling)。

如要插入非图形化字符，先输入 C-q (quoted-insert)

- 输入 DEL。C-q 后，紧接着输入 <DEL>。
- 输入 Unicode。C-q 1 0 1 B 显示 AB。

read-quoted-char-radix 控制基数，如果为 10 表示十进制，如果为 16 表示十六进制。

Unicode 字符还可以通过 C-x 8 命令插入，C-x 8 C-h 查看具体插入 Unicode 字符的命令。例如，C-x 8 \$ 插入字符 ♂。或者 C-x 8 <RET> 会列出所有 Unicode 可用字符。例如，输入 lambda，找到对应命令 Greek Small Letter Lambda 就能插入 λ。

#### 5.1.2 移动光标 (Move Point)

我使用 Evil-mode 所以不太用这些操作。

#### 5.1.3 删除 (Erasing)

Emacs	Function	Evil
<DEL>	delete-forward-char	x
<BACKSPACE>	delete-backward-char	X
C-d	delete-char	x
C-k	kill-line	dd
M-d	kill-word	D

#### 5.1.4 基本撤销 (Basic Undo)

Emacs	Function	Evil
C-x u	undo	u
M-x _	redo	

#### 5.1.5 文件 (Files)

Emacs	Function	Evil
C-x C-f	find-file	
C-x C-s	save-buffer	

#### 5.1.6 帮助 (Help)

简单，直接 C-h 即可。



## 6.2 Document Structure

### 6.2.1 Headlines

local visible cycling	<tab>
global visible cycling	<backtab>
move up/down	<M-up> / <M-down>

## 6.3 ToDo Items

1. Basic

## 6.4 Datetimes

ToDo items 可以标记 date 和 time，在 org 中称之为 timestamp。

### 6.4.1 Timestamps

时间戳分类

1. 普通时间戳；事件；约会 只分配 date/time 形式的时间戳，在显示 agenda 时，只显示 date。
  - (a) 周六看电影 <2019-11-30 Sat>
  - (b) 周四吃晚餐 <2019-11-28 Thu>
2. 规律重复出现的时间戳 如每天下午 5 点跑步，每周六去公园玩，每年的生日等。**d** 表示 day，**w** 表示 week，**m** 表示 month。
3. 用 sexp 表示复杂日期 牵涉到 LISP 语言中的 S-exp 表达式，暂不考虑。可以用来安排课程表。
4. 指定 Time/Date 范围 由 -- 连接的两个日期表示范围。
  - (a) 论文答辩 <2019-12-10 Tue>--<2019-12-13 Fri>
5. 非激活日期 由 [ ] 包含的日期，并不会出现在 agenda 中。
  - (a) 和朋友聚餐 [2019-11-28 Thu]

### 6.4.2 创建时间戳

C-c .	插入日期/连续日期
C-c !	插入非激活日期
C-c <	插入当前日期
C-c C-o	列出光标下日期/日期范围中的 agenda
<S-right> <S-left>	调整月份
<S-up> <S-down>	调整天数
C-c C-y	计算有多少天



### 6.4.3 Deadlines 和 Scheduling

时间戳可配置特殊关键字帮助进行计划，例如 Deadlines 和 Scheduling。

使用 C-c / 会出现 sparse tree。

1. DEADLINE 所有 DEADLINE 日期之前计划的事件都会显示在 agenda 中,org-deadline-warning-days 指定 DEADLINE 日期之前几天发出警告，直到将事件标志为 DONE 才会停止提示警告。

#### (a) 完成课程教案编写工作

DEADLINE:<2019-11-30 Sat>

第一次编辑:[bldb:Ford Prefect]

使用 -2d 表示提前 2 天发出警告提示。

#### (b) 与张总见面约谈

DEADLINE:<2019-11-30 Sat -2d>

重复某个 Deadlines，使用 +1m 表示每月重复一次，~-2d~ 表示提前 2 天提示。当本月任务完成后，再次使用 C-c C-t 不但能标记本月任务已经完成，而且还能启用下月任务。凡是标记为 DONE 的任务不再出现在 agenda 中。

#### (c) 交房租

DEADLINE: <2019-08-15 Thu +1m -2d>

:PROPERTIES:

:LAST\_REPEAT: [2019-11-28 Thu 15:27]

:END:

#### (d) 与 Scott 老师视频通话

SCHEDULED: <2019-10-25 Fri ++1w>

:PROPERTIES:

:LAST\_REPEAT: [2019-11-28 Thu 15:26]

:END:

2. SCHEDULED 计划何时开始某项新任务。如果延迟两天才开始该任务，则显示为 2x。如果计划推迟 3 天，则在时间上使用 -3d 表示。

SCHEDULED 经常会被误解。例如，与某人约会是 appointment，使用简单的计划即可。一旦使用 SCHEDULED，则表示在该任务在指定日期才会在时间线中出现。

#### (a) 去新加坡旅行

SCHEDULED:<2019-11-24 Sun --2d>

## 7 Timer Example

### 7.1 Clock column view example

### 7.2 Effort estimate example

## 8 Orgmode (Studing)

### 8.1 E01S01 : Headlines & outline mode

**CLOSED:** [2020-02-10 Mon 21:25]

Local expand/collapse cycle	<tab>
Global expand/collapse cycle	<backtab>
Increase level	<alt-right>
Decrease level	<alt-left>
Move up	<M-up>
Move down	<M-down>

### 8.2 E01S03 : Schedule, Deadlines & Agenda views

Schedule done	C-c C-s
Deadlines defined	C-c C-d
Agenda switch	C-a a
Follow mode	S-f
Move forward	f
Move backward	b

### 8.3 E01S04 : Repeating tasks

regular every week (d=day w=week m=month y=year)	+1w
repeat every week, definitely in the future	++1w
Next 4 week after the task is DONE	+.4w

If repeated event was setting by SCHEDULED C-c C-t , you could press C-c C-t again to change current SCHEDULED event to state DONE and restart a new SCHEDULED event.

#### 8.3.1 规律重复出现的时间戳

如每天下午 5 点跑步，每周六去公园玩，每年的生日等。**d** 表示 day，**w** 表示 week，**m** 表示 month。

### 8.4 E01S05 : Checklists [2/4]

☐ This is checklist

- ☐ New checklist by <M-S-return>
- ☒ You can see the completion of the checklist with [0/0] or [0%] on headlines.
- ☒ Checkbox toggled with C-c C-c

## 8.5 E02S01 : Tags

Tags can be added to headlines and are often used as a way to mark GTD contexts.

Tags can be predefined with a line `#+TAGS:PHONE(o) . . .` at the beginning of the file. Use C-c C-q to assign a tags for a headline. Use <tab> to enter new tags instead of predefined tags. Clear tags by SPC .

## 8.6 E02S02 : Agenda view (advanced)

Timeline for current buffer	L
List all TODO entries	t
Entries with a special TODO Keywords	T
Match tags/propoerties/TODO keywords	M
Like before, but only TODO keywords	M
Search for keywords	s

## 8.7 E02S03 : Customized Agenda view

for customizing agenda command

## 8.8 E02S04 : Drawers,Logging & quick notes

- Note taken on *[2020-02-12 Wed 10:14]*  
test quick notes

I want to add quick notes. If you want `:LOGBOOK:` appeared in the body, you must customize `org-log-into-drawer`.

We could create **DRAWER** by C-c C-x d .

- This is the usually **hidden** content of drawer
- The drawer ends with the line that says `:END:`
- Reserved drawer names are e.g. `:LOGBOOK:` or `:PROPERTIES:`

`:LOGBOOK:` don't display in the ouput html files!?

This is the source orgfiles

`:LOGBOOK:`

- Note taken on [2020-02-12 Wed 10:25] \\

After customize `~org-log-into-drawer~`, I could get quick notes into body.

- Note taken on [2020-02-12 Wed 10:09] \\

This is a quick notes using `@@html:<kbd>@@ C-c C-z @@html:</kbd>@@` . That's good!  
:END:

- Note taken on *[2020-02-12 Wed 15:36]*  
You can use `#+OPTIONS: d:t` to show drawer.
- Note taken on *[2020-02-12 Wed 10:25]*  
After customize `org-log-into-drawer`, I could get quick notes into body.
- Note taken on *[2020-02-12 Wed 10:09]*  
This is a quick notes using `C-c C-z` . That's good!

## 8.9 E02S05 : Archiving

Finished tasks collected up your org file. Define a global archives file with `#+ARCHIVE: myarchive.org:..`  
Archive an entry with `C-c C-x C-a` . Archive a subtree with `C-c C-x C-s` .

When you complete you task, you can move out of your DONE tasks to archives files.

## 8.10 E03S01 : Automatic logging of status changes

automatic logging of status changes for complicated tasks. I think I will never use this feature.

## 8.11 E03S02 : Splitting your system up to several files

See in the future.

## 8.12 E03S03 : The first capture template(s)

Goal: Capturing tasks, ideas and whatever you want!

Use `SPC C c` to open capture template.

### 8.13 E03S04 : The -PROPERTIES - drawer

### 8.14 E03S05 : Archiving to different files

### 8.15 E04S01 : Ordered tasks

### 8.16 E04S02 : Timers

---

Clocking in	C-c C-x C-i
Clocking out	C-c C-x C-o
Restart a clock	C-c C-x C-x
Jump to Clocked task	C-c C-x C-j
Cancel a Clock in	C-c C-x C-q
Show times	C-c C-x C-d
Recompute time	C-c C-c
Start countdown timer	C-c C-x ;
Pause a timer or continue	C-c C-x ,
Insert current timer	C-c C-x .
Start a relative timer	C-c C-x 0
Calc & display spend time	C-c C-x C-d
Enter Column view	C-c C-x C-c
Leave column view	Q

---

### 8.17 E04S03 : Clocking (aka time tracking)

#### 8.17.1 Football notes

You'll start relative timer by C-c C-x 0 . The timer will be tick from 00:00:01, 00:00:02, ... , 00:01:08, ....

Suppose team A get a score at sometimes, press C-c C-x . to get current relative time and insert below current cursor.

**0:04:31** Team A scored a goal 1:0 .

**0:05:26** [When press O to get a newline, insert current timer automatically.] Team B scored a goal 1:1.

[If you go out to get a drink, you could press C-c C-x , to pause timer. Repeat such combination keys when coming back from outside.]

- Stop timer by C-c C-x \_

#### 8.17.2 Clock Tracking example

##### 1. Task A

- Some task that we need to know how much time it take.

Use C-c C-x TAB to insert timestamp, Use C-c C-x C-j to jump to the last insert timestamp, Use C-c C-x C-o to close timestamp.

## 2. Task B

- Some other task to clock.

Use C-c C-c to recompute timer. Use C-c C-x C-d to calculate spend time. Use C-c C-c to cancel display total spend time.

### 8.18 E04S04 : Column view

`#+COLUMNS: %7TODO(To Do) %40ITEM(TASK) %TAGS(Tags) %6CLOCKSUM(Clock)`

Enter Column view	C-c C-x C-c
Leave column view	Q

### 8.19 E04S05 : Effort estimates

Goal: Estimate the effort that your task will take.

Recipe: Effort are properties stored in `:Effort:`. Easy setup: Define `#+PROPERTIES: Effort_ALL`

Add `#+COLUMNS: %8Effort(Effort){:}` in column view. `{:}` means sum up times.

Show column view	C-c C-x C-c
increase effort	<S-right>
decrease effort	<S-left>
Leave column view	Q

### 8.20 E05S01 : Linking (internal)

Links is `[[target] [description]]`.

Edit link	C-c C-l
Follow the link	C-c C-o
Return to previous link	C-c &

- This is a headline link to [HEADLINE link](#). Use C-c C-l to edit internal link.

`[[E01S02 : ToDo Keywords]]`

- This is a `#+NAME:timerlink` link to `[[timerlink] [NAME Link to E04S02 Timers]]`.
- This is a `:CUSTOM_ID:` link to [CUSTOM\\_ID LINK](#).
- Use `<<<target>>>` to create links on the fly.

## 8.21 E05S02 : Linking (external)

## 8.22 E05S03 : Attachments

## 8.23 E05S04 : Priorities

## 8.24 E05S05 : Tables

## 8.25 E06S01 : Exporting

- Goal: export files into HTML and PDF
- Key : C-c C-e
- Customization: Setting `org-file-apps` determines the application to open files.

### 8.25.1 export chinese pdf ?

[\*IMPORTANT\*] I have been set how to open html files with `qutebrowser` and how to open pdf files with `zathura`.

Add a line with `#+OPTIONS:` to fine tune the output, `t` is true.

Show the contents of drawer	<code>d:t</code>
Preserve line breaks	<code>\n:t</code>
Export planing information	<code>p:t</code>
Include TODO keywords into exported text	<code>todo:t</code>

## 8.26 E06S02 : Advanced exporting

- M-x `customize-group` then input `org-export`
- add `+AUTHOR` and `+TITLE`
- There has many choices to export pdf!

```
# for beamer
#+SUBTITLE: ORGMODE EXPLAINED
#+BEAMER_THEME: Berlin
#+BEAMER_FONT_THEME: professionalfonts
```

### **8.27 E06S03 : Publishing**

### **8.28 E06S04 : Dynamic blocks**

### **8.29 E06S05 : Tracking habits**

### **8.30 E07S01 : Bulk agenda actions**

### **8.31 E07S02 : Presenting my system**

### **8.32 E07S03 : Google Calendar integration**

### **8.33 E07S04 : Source code in OrgMode**

### **8.34 E07S05 :**

## **9 Use Emacs**

### **9.1 62 : Magit**

I must learn git before using Magit.

## **10 Git**

### **10.1 Git Basic**

#### **10.1.1 in master**

1. `git init`
2. `git config --global user.name 'AiPick'`
3. `git config --global user.email '2585957571@qq.com'`
4. `git add *.*` all files **NOT** including subdir `git add .` add all file include subdir
5. `git status`
6. `git rm --cached [uncached filename]`
7. `git commit -m "add readme.txt"`
8. `touch .gitignore` ignore these files

#### **10.1.2 switch to branch**

1. `git branch mybranch`
2. `git checkout mybranch`
3. `touch branch.txt`



4. `git add .`
5. `git commit -m "branch changed"`

### 10.1.3 switch back to master and merge

1. `git checkout master`
2. `git merge mybranch`

### 10.1.4 git to remote

1. `git remote add origin https://github.com/AiPick/notes.git`
2. `git remote`
3. `git push -u origin master` input username & password
4. refresh github pages to get pushed files and dirs
5. `touch newtest.txt`
6. `git add .`
7. `git commit -m "test with newtest.txt which pushed to remote"`
8. `git push` before refresh github pages to get `newtest.txt` appeared in remote repo.

### 10.1.5 get newest repo

1. `git pull`

## 10.2 magit

1. SPC g s magit status
2. committed file s
3. committed to repo c
4. use , , to actually commit changed files.
5. use p to push local repo to remote repo.
6. use F to pull from remote repo.
7. use e to solve conflict in Ediff. use N & P to navigate between the Ediff. use A & B to choice what you want to reserve code.
8. SPC g b is blame state to show different author comment and revised code on the timeline. You could press b to backtrace in the history.

9. use f to fetch repo files<sup>[1]</sup> .
- 10.<sup>[2]</sup>

## 参考文献

- [1] RÉMY D. Using, understanding, and unraveling the OCaml language from practice to theory and vice versa[C]// International Summer School on Applied Semantics. Springer. [S.l.]: [s.n.], 2000: 413–536 (引用页: 18).
- [2] KOSBA A, MILLER A, SHI E, et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts[C]// 2016 IEEE symposium on security and privacy (SP). IEEE. [S.l.]: [s.n.], 2016: 839–858 (引用页: 18).