Learn Emacs through org-mode

Linyxus[1]

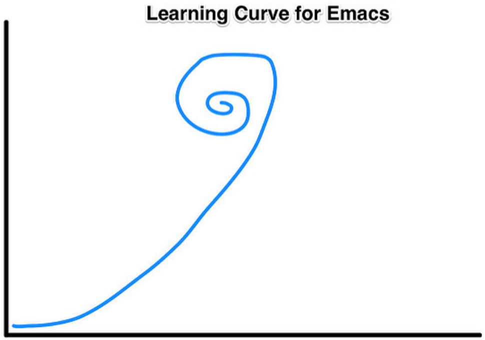
Dec 2018

[1] linyxus00@gmail.com

Outline

Learning curve

Emacs may be best known for its learning curve:



History[2]

A brief list:

- 1970s, in Artificial Intelligence Laboratory at MIT, TECO
- 1976, by Stallman, the first Emacs("Editor MACroS")
- 1978, by Bernard Greenberg, MulticsEmacs, **introducing MacLisp**
- 1981, the first Emacs to run on Linux, Gosling Emacs
- 1984, by Stallman, **GNU Emacs**

[2]according to EmacsWiki

What a excellent editor is like

- Highly extensible (Emacs can do everthing!)
- FLexible (freely define your own key bindings)
- Portable (bring your Emacs everywhere)
- Compatible (GUI && Terminal)
- Macros

Emacs deserves your efforts

- It will never be out of date.
- Be used in a wide range.
  - Programming
  - Documenting
  - Mailing
  - IRC
  - Playing games
  - ...
- It's really powerful.

Aim of this lecture

Find your passion

The best way to learn Emacs is to use it. Find out how Emacs can help you.

Give you a overview of Emacs

- What can Emacs do?
- Where to get started?

Configure Emacs

Step I: Find your .emacs.d

run the following command: C-h v user-emacs-directory

- Linux & macOS: usually at ~/.emacs.d
- Windows: depends on your HOME variable

Step II: Download the configuration of Purcell

Github: `https://github.com/purcell/emacs.d`
Or open Github.com and search ‘emacs.d’.

Step III: Place the config & restart your Emacs

- Override your .emacs.d with Purcell’s
- Restart your Emacs

Keys in Emacs

Abbr	Meaning
S	Shift
C	Ctrl
M	Meta[3]

Here are some examples:

- C-f and M-f
- C-a a and C-a C-a

[2]Meta -> Alt or Esc

Basic mouse movements

Key	Function
C-v	page down
M-v	page up
C-f	one character *f*orward
M-f	one word *f*orward
C-b	one character *b*ackward
M-b	one word *b*ackward
C-n	*n*ext line
C-p	*p*revious line
M-> and M-<	to the end / beginning of the buffer

Of course, there are more movements.
But the ones listed above are enough for now.

What is actually happenning?

- Hit the keys == call the function bound to them
- Almost everything you do in Emacs is just calling a function.
- This means you can fully customize the behavior of your Emacs.

File, buffer and window

File

- When you **find**[4] a file: file -> buffer

Buffer

- No changes are applied to your file until you save the buffer.
- Not every buffer has a corresponding file.
- A backup is created when you save a buffer to a existing file.

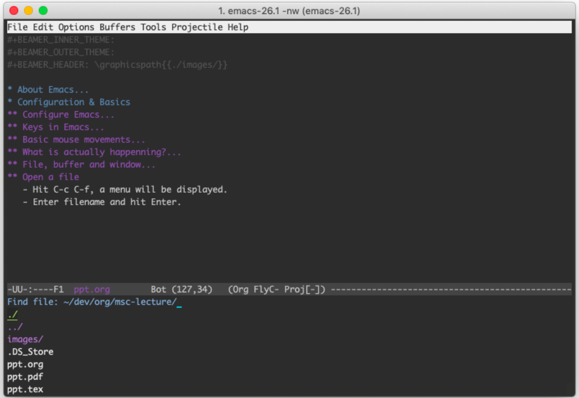
Window

- Each window has a corresponding buffer.
- Multi-window is possible.

[4] find ~ open

Open a file

- Hit C-c C-f, a menu will be displayed.
- Enter filename and hit Enter.

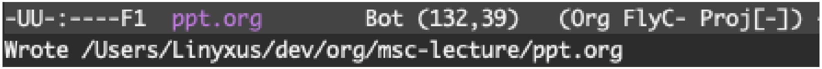

1. emacs-26.1 -nw (emacs-26.1)
File Edit Options Buffers Tools Projectile Help

```
#+BEAMER_INNER_THEME:
#+BEAMER_OUTER_THEME:
#+BEAMER_HEADER: \graphicspath{{./images/}}

* About Emacs...
* Configuration & Basics
** Configure Emacs...
** Keys in Emacs...
** Basic mouse movements...
** What is actually happenning?...
** File, buffer and window...
** Open a file
   - Hit C-c C-f, a menu will be displayed.
   - Enter filename and hit Enter.

-UU-:----F1  ppt.org       Bot (127,34)   (Org FlyC- Proj[-]) ------------------------------------------
Find file: ~/dev/org/msc-lecture/_
./
../
images/
.DS_Store
ppt.org
ppt.pdf
ppt.tex
```

Save, kill or switch to a buffer I

Save

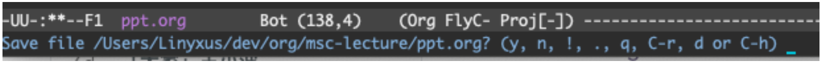
Hit C-x C-s to save current buffer.



```
-UU-:----F1  ppt.org          Bot (132,39)   (Org FlyC- Proj[-])
Wrote /Users/Linyxus/dev/org/msc-lecture/ppt.org
```

Hit C-x s to save all.

Key	Action
y / n	save / not save this file (asked one by one)
!	save all

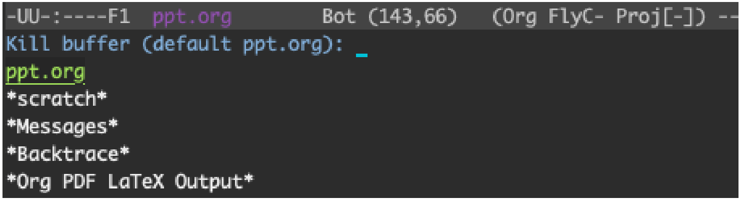


```
-UU-:**--F1  ppt.org          Bot (138,4)    (Org FlyC- Proj[-]) -------------------------
Save file /Users/Linyxus/dev/org/msc-lecture/ppt.org? (y, n, !, ., q, C-r, d or C-h) _
```

Save, kill or switch to a buffer II

Kill

Hit C-x k, enter the name of the buffer to kill and hit Enter.

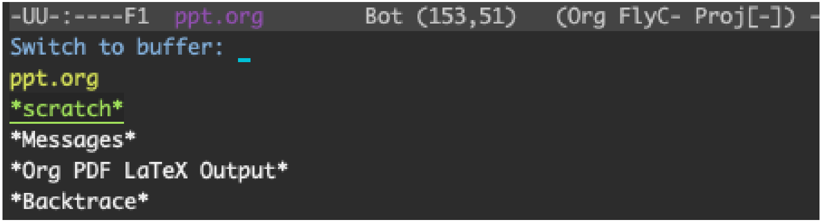


```
-UU-:----F1  ppt.org        Bot (143,66)   (Org FlyC- Proj[-]) --
Kill buffer (default ppt.org): _
ppt.org
*scratch*
*Messages*
*Backtrace*
*Org PDF LaTeX Output*
```

Switch to another buffer

Save, kill or switch to a buffer III

Let the **current** window display another buffer.
Hit C-x b, enter the buffer name and hit Enter.



```
-UU-:----F1  ppt.org        Bot (153,51)   (Org FlyC- Proj[-]) -
Switch to buffer: _
ppt.org
*scratch*
*Messages*
*Org PDF LaTeX Output*
*Backtrace*
```

Window management

Key	Function
C-x 3	Split current window
C-x 0	Close current window
C-x 1	Close all windows but for current one

org-mode as a GTD tool I

Well, but what is GTD?

- GTD == Get Things Done
- Getting Things Done is a time management method, described in the book of the same title by productivity consultant David Allen. The method is often referred to as GTD.[5]
- GTD ≈ Methods, tools centered around your Todo List.
- a GTD tool > a todo list
  - capture
  - calendar: schedule & deadline
  - categories & projects & states

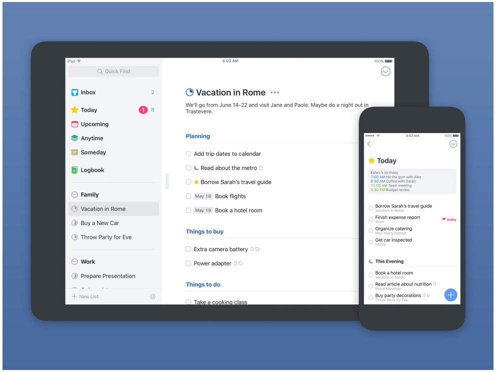
org-mode as a GTD tool II

Of course, org-mode can do much more than what an ordinary GTD tool can do.

- org-mode as a GTD tool > ordinary GTD tools
- org-mode > a GTD tool
- Emacs > org-mode

[5] https://en.wikipedia.org/wiki/Getting_Things_Done

Things 3: a typical GTD tool I



Vacation in Rome

We'll go from June 14-22 and visit Jane and Paolo. Maybe do a night out in Trastevere.

Planning

- Add trip dates to calendar
- Read about the metro
- Borrow Sarah's travel guide
- May 18 Book flights
- May 19 Book a hotel room

Things to buy

- Extra camera battery
- Power adapter

Things to do

- Take a cooking class

Today

Marc's birthday
7:00 AM Hit the gym with Alex
8:30 AM Coffee with Sarah
11:00 AM Team meeting
3:30 PM Budget review

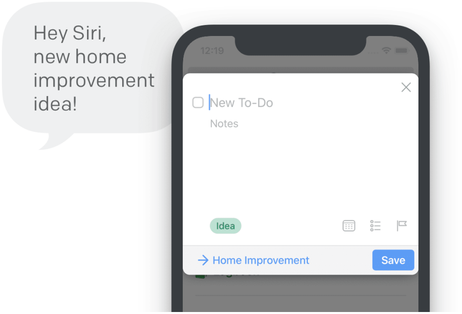
- Borrow Sarah's travel guide — Vacation in Rome
- Finish expense report — Work — today
- Organize catering — Plan Yearly Retreat
- Get car inspected — Family

This Evening

- Book a hotel room — Vacation in Rome
- Read article about nutrition — Run a Marathon
- Buy party decorations — Throw Party for Eve

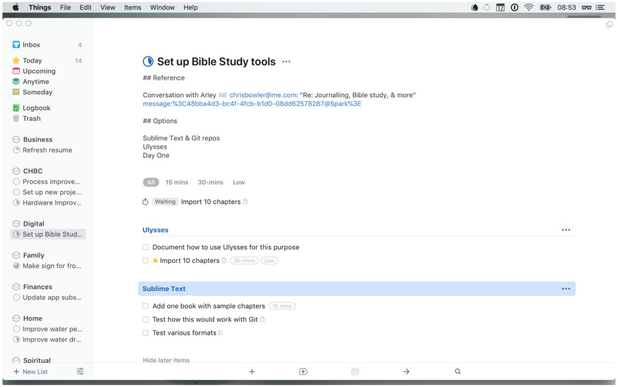
Things 3: a typical GTD tool II

capture your matters & ideas



Things 3: a typical GTD tool III

project planning & tags



Set up Bible Study tools ···

Reference

Conversation with Arley chrisbowler@me.com: "Re: Journalling, Bible study, & more"
message:%3C48bba4d3-bc4f-4fcb-b1d0-08dd62578287@Spark%3E

Options

Sublime Text & Git repos
Ulysses
Day One

All 15 mins 30-mins Low

Waiting Import 10 chapters

Ulysses

- Document how to use Ulysses for this purpose
- Import 10 chapters 30-mins Low

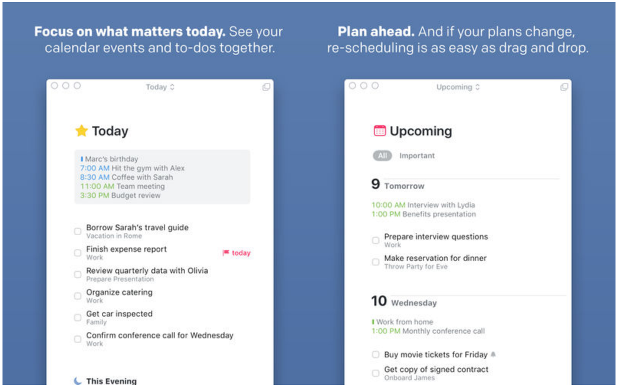
Sublime Text

- Add one book with sample chapters 15 mins
- Test how this would work with Git
- Test various formats

Hide later items

Things 3: a typical GTD tool IV

schedule & agenda



Focus on what matters today. See your calendar events and to-dos together.

Today

Today

Marc's birthday
7:00 AM Hit the gym with Alex
8:30 AM Coffee with Sarah
11:00 AM Team meeting
3:30 PM Budget review

- Borrow Sarah's travel guide
  Vacation in Rome
- Finish expense report — today
  Work
- Review quarterly data with Olivia
  Prepare Presentation
- Organize catering
  Work
- Get car inspected
  Family
- Confirm conference call for Wednesday
  Work

This Evening

Plan ahead. And if your plans change, re-scheduling is as easy as drag and drop.

Upcoming

Upcoming

All Important

9 Tomorrow

10:00 AM Interview with Lydia
1:00 PM Benefits presentation

- Prepare interview questions
  Work
- Make reservation for dinner
  Throw Party for Eve

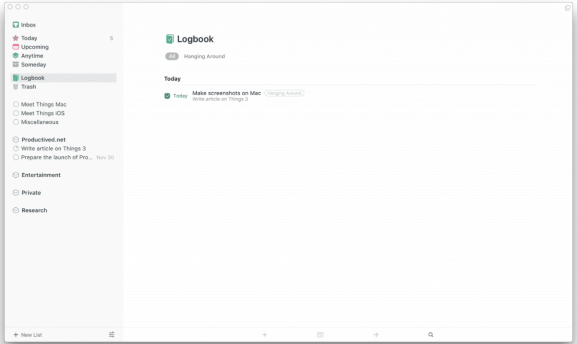
10 Wednesday

Work from home
1:00 PM Monthly conference call

- Buy movie tickets for Friday
- Get copy of signed contract
  Onboard James

Things 3: a typical GTD tool V

archive after completion



org-mode can do it!

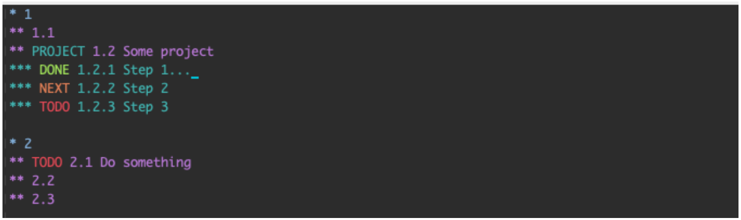
org-mode can do all what Things 3 can do! **(actually more)**
Despite the drawbacks:

- No notifications, only beeps
- Cannot be asynced everywhere (Things3: iPad, macOS, iOS)

Where to use org-mode: In your projects!

- essay
- homework
- programming
- or whatever. . .

Your first .org file

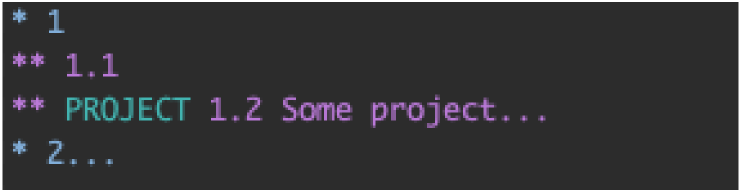


```
* 1
** 1.1
** PROJECT 1.2 Some project
*** DONE 1.2.1 Step 1..._
*** NEXT 1.2.2 Step 2
*** TODO 1.2.3 Step 3

* 2
** TODO 2.1 Do something
** 2.2
** 2.3
```

- Headings: *
- Todo keywords: PROJECT, TODO, NEXT

Cycle through

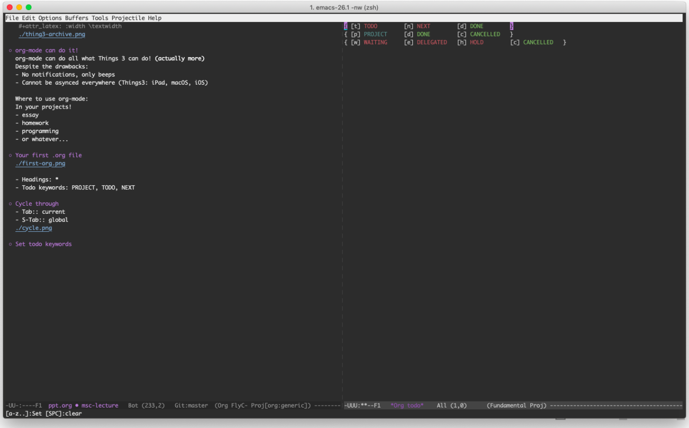
- Tab:: current
- S-Tab:: global



```
* 1
** 1.1
** PROJECT 1.2 Some project...
* 2...
```

Set todo keywords

- Of course, you can type in the keywords manually.
- Shortcut: C-t t



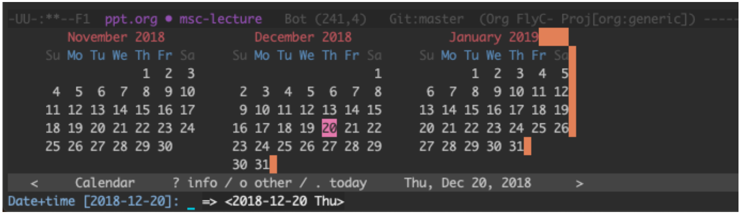
```
  #+attr_latex: :width \textwidth
  ./thing3-archive.png

○ org-mode can do it!
  org-mode can do all what Things 3 can do! (actually more)
  Despite the drawbacks:
  - No notifications, only beeps
  - Cannot be asynced everywhere (Things3: iPad, macOS, iOS)

  Where to use org-mode:
  In your projects!
  - essay
  - homework
  - programming
  - or whatever...

○ Your first .org file
  ./first-org.png

  - Headings: *
  - Todo keywords: PROJECT, TODO, NEXT

○ Cycle through
  - Tab:: current
  - S-Tab:: global
  ./cycle.png

○ Set todo keywords
```

```
{ [t] TODO        [n] NEXT        [d] DONE        }
{ [p] PROJECT     [d] DONE        [c] CANCELLED   }
{ [w] WAITING     [e] DELEGATED   [h] HOLD        [c] CANCELLED   }
```

```
-UU-:----F1  ppt.org ● msc-lecture   Bot (233,2)   Git:master  (Org FlyC- Proj[org:generic]) -------- -UUU:**--F1   *Org todo*   All (1,0)     (Fundamental Proj) --------------------------------------
[a-z..]:Set [SPC]:clear
```

Schedule everything! I

set scheduled time
Press C-c C-s, choose the time.



```
-UU-:**--F1  ppt.org ● msc-lecture   Bot (241,4)   Git:master  (Org FlyC- Proj[org:generic]) ----
        November 2018            December 2018            January 2019
     Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa
                  1  2  3                       1            1  2  3  4  5
      4  5  6  7  8  9 10      2  3  4  5  6  7  8      6  7  8  9 10 11 12
     11 12 13 14 15 16 17      9 10 11 12 13 14 15     13 14 15 16 17 18 19
     18 19 20 21 22 23 24     16 17 18 19 20 21 22     20 21 22 23 24 25 26
     25 26 27 28 29 30        23 24 25 26 27 28 29     27 28 29 30 31
                              30 31
  <     Calendar    ? info / o other / . today     Thu, Dec 20, 2018      >
Date+time [2018-12-20]: _  => <2018-12-20 Thu>
```

set deadline
Press C-c C-d. Similar.

Schedule everything! II

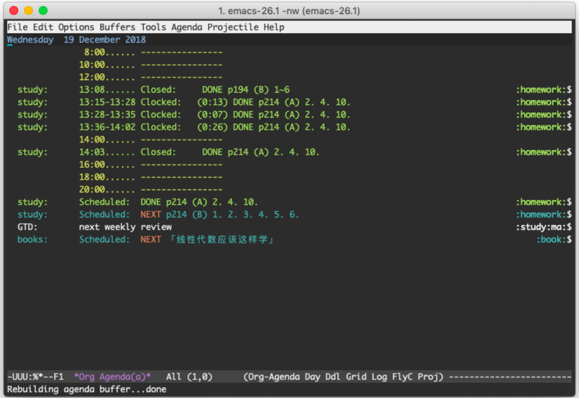
Possible time strings

	Today
+1	Tomorrow
+1Fri	Next Friday
+1w	Equal to +7
+1m	Next month
+1y	Next year
20:00	20:00 Today
+4 20:00	4 days later, 20:00

Agenda I

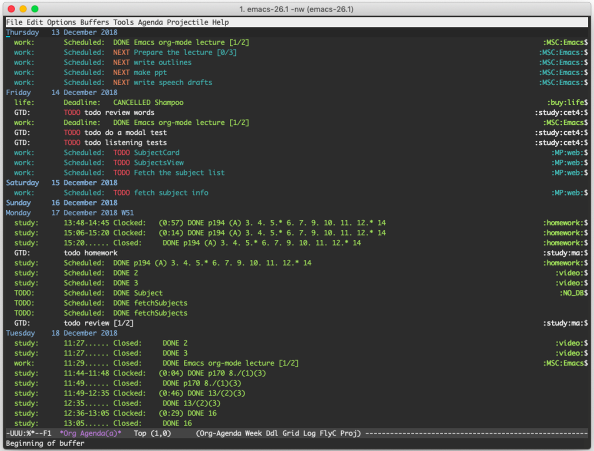
- Give an overview of your matters
- Arrange / Modify your todos (scattered in many files) in one view
- Make up your plan

Agenda II


1. emacs-26.1 -nw (emacs-26.1)
File Edit Options Buffers Tools Agenda Projectile Help

```
Wednesday  19 December 2018
                8:00...... ----------------
               10:00...... ----------------
               12:00...... ----------------
  study:       13:08...... Closed:     DONE p194 (B) 1~6                                       :homework:$
  study:       13:15-13:28 Clocked:    (0:13) DONE p214 (A) 2. 4. 10.                          :homework:$
  study:       13:28-13:35 Clocked:    (0:07) DONE p214 (A) 2. 4. 10.                          :homework:$
  study:       13:36-14:02 Clocked:    (0:26) DONE p214 (A) 2. 4. 10.                          :homework:$
               14:00...... ----------------
  study:       14:03...... Closed:     DONE p214 (A) 2. 4. 10.                                 :homework:$
               16:00...... ----------------
               18:00...... ----------------
               20:00...... ----------------
  study:       Scheduled:  DONE p214 (A) 2. 4. 10.                                             :homework:$
  study:       Scheduled:  NEXT p214 (B) 1. 2. 3. 4. 5. 6.                                     :homework:$
  GTD:         next weekly review                                                              :study:ma:$
  books:       Scheduled:  NEXT 「线性代数应该这样学」                                                :book:$

-UUU:%*--F1  *Org Agenda(a)*   All (1,0)      (Org-Agenda Day Ddl Grid Log FlyC Proj) -----------------------
Rebuilding agenda buffer...done
```

Agenda III



1. emacs-26.1 -nw (emacs-26.1)

```
File Edit Options Buffers Tools Agenda Projectile Help
Thursday   13 December 2018
  work:       Scheduled:  DONE Emacs org-mode lecture [1/2]                                                                       :MSC:Emacs$
  work:       Scheduled:  NEXT Prepare the lecture [0/3]                                                                        :MSC:Emacs:$
  work:       Scheduled:  NEXT write outlines                                                                                   :MSC:Emacs:$
  work:       Scheduled:  NEXT make ppt                                                                                         :MSC:Emacs:$
  work:       Scheduled:  NEXT write speech drafts                                                                              :MSC:Emacs:$
Friday     14 December 2018
  life:       Deadline:   CANCELLED Shampoo                                                                                       :buy:life$
  GTD:        TODO todo review words                                                                                         :study:cet4:$
  work:       Deadline:   DONE Emacs org-mode lecture [1/2]                                                                       :MSC:Emacs$
  GTD:        TODO todo do a modal test                                                                                      :study:cet4:$
  GTD:        TODO todo listening tests                                                                                      :study:cet4:$
  work:       Scheduled:  TODO SubjectCard                                                                                       :MP:web:$
  work:       Scheduled:  TODO SubjectsView                                                                                      :MP:web:$
  work:       Scheduled:  TODO Fetch the subject list                                                                            :MP:web:$
Saturday   15 December 2018
  work:       Scheduled:  TODO fetch subject info                                                                                :MP:web:$
Sunday     16 December 2018
Monday     17 December 2018 W51
  study:      13:48-14:45 Clocked:   (0:57) DONE p194 (A) 3. 4. 5.* 6. 7. 9. 10. 11. 12.* 14                                     :homework:$
  study:      15:06-15:20 Clocked:   (0:14) DONE p194 (A) 3. 4. 5.* 6. 7. 9. 10. 11. 12.* 14                                     :homework:$
  study:      15:20...... Closed:     DONE p194 (A) 3. 4. 5.* 6. 7. 9. 10. 11. 12.* 14                                           :homework:$
  GTD:        todo homework                                                                                                      :study:ma:$
  study:      Scheduled:  DONE p194 (A) 3. 4. 5.* 6. 7. 9. 10. 11. 12.* 14                                                       :homework:$
  study:      Scheduled:  DONE 2                                                                                                    :video:$
  study:      Scheduled:  DONE 3                                                                                                    :video:$
  TODO:       Scheduled:  DONE Subject                                                                                               :NO_DB$
  TODO:       Scheduled:  DONE fetchSubjects
  TODO:       Scheduled:  DONE fetchSubjects
  GTD:        todo review [1/2]                                                                                                  :study:ma:$
Tuesday    18 December 2018
  study:      11:27...... Closed:     DONE 2                                                                                        :video:$
  study:      11:27...... Closed:     DONE 3                                                                                        :video:$
  work:       11:29...... Closed:     DONE Emacs org-mode lecture [1/2]                                                           :MSC:Emacs$
  study:      11:44-11:48 Clocked:   (0:04) DONE p170 8./(1)(3)
  study:      11:49...... Closed:     DONE p170 8./(1)(3)
  study:      11:49-12:35 Clocked:   (0:46) DONE 13/(2)(3)
  study:      12:35...... Closed:     DONE 13/(2)(3)
  study:      12:36-13:05 Clocked:   (0:29) DONE 16
  study:      13:05...... Closed:     DONE 16
-UUU:%*--F1  *Org Agenda(a)*   Top (1,0)      (Org-Agenda Week Ddl Grid Log FlyC Proj) -----------------------------------------------------
Beginning of buffer
```

Agenda commands

v [d/w/m/y]	Switch to day/week/month/year view
f	forward
b	backward
.	current
t	set todo keywords
C-c C-s	schedule
C-c C-d	set deadline
F	following mode
SPC	goto todo (cursor unmoved)
TAB	goto todo (move cursor in the file)
q	quit
r	rebuild buffer
/	filter

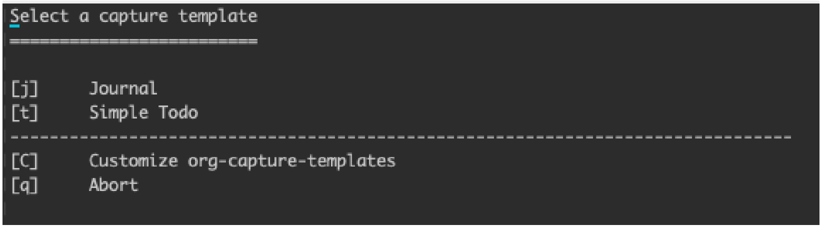
Capture ideas

- An idea / matter comes to your when you are busy with other things?
- Take a quick note?

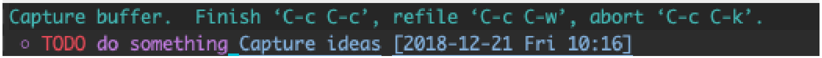
Capture ideas

Step 1: Capture it!

1. Press C-c c and 'select template.
2. Put down your ideas and press C-c C-c.
3. Done!



```
Select a capture template
=========================

[j]     Journal
[t]     Simple Todo
-------------------------------------------------------------------------------
[C]     Customize org-capture-templates
[q]     Abort
```



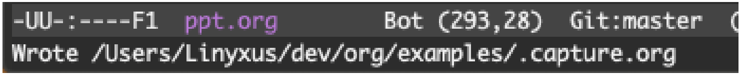
```
Capture buffer.  Finish ‘C-c C-c’, refile ‘C-c C-w’, abort ‘C-c C-k’.
 ○ TODO do something Capture ideas [2018-12-21 Fri 10:16]
```

Capture ideas I

Step 2: Review && Refile

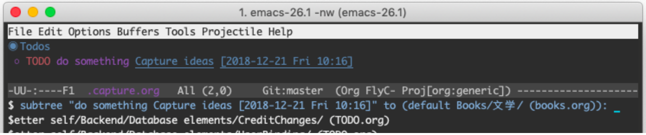
Capture ideas II

When C-c C-c is pressed, your entry will be saved to a specific file.



```
-UU-:----F1  ppt.org        Bot (293,28)  Git:master  (
Wrote /Users/Linyxus/dev/org/examples/.capture.org
```

1. Open the file.
2. Move the cursor to the entry, and press C-c C-w.
3. Select where you want to refile your todo to.
4. Done!


1. emacs-26.1 -nw (emacs-26.1)
File Edit Options Buffers Tools Projectile Help

```
◉ Todos
  ○ TODO do something Capture ideas [2018-12-21 Fri 10:16]

-UU-:----F1  .capture.org   All (2,0)    Git:master  (Org FlyC- Proj[org:generic]) -------------------
$ subtree "do something Capture ideas [2018-12-21 Fri 10:16]" to (default Books/文学/ (books.org)): _
$etter self/Backend/Database elements/CreditChanges/ (TODO.org)
```

Advantages of org-mode (compared to Things 3)

- More accurate daily agenda (with exact timetable)
- Mixing notes with todos
- Easy export & share

org-mode can do MORE!

- custom capture (capture templates)
- links (links everything!)
- attachments
- clock & timer
- notes taking
- various properties

But org-mode is not only a GTD tool!

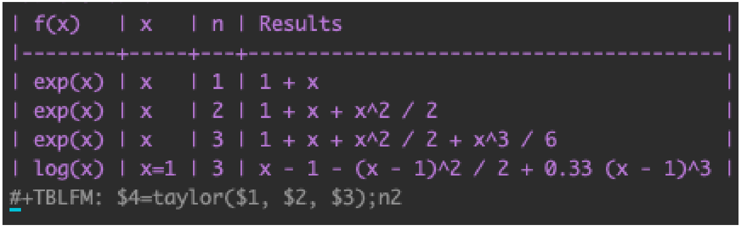
An overview of what can org-mode do: the Google Tech Talk given by org-mode creator Dominik[6]

- write documents (export to txt, html, LaTeX, pdf...)[7]
- data processing
- embedded in programming
- ...

[6] `https://www.youtube.com/watch?v=oJTwQvgfgMM`
[7] Actually, this ppt is produced by org-mode.

Table & calc

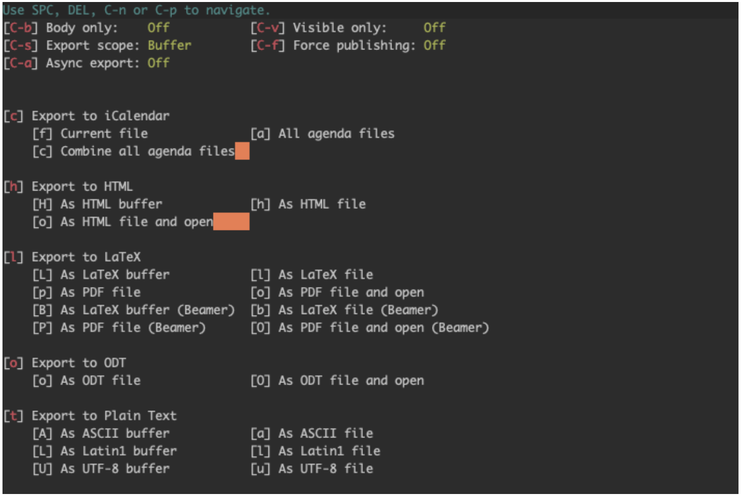
f(x)	x	n	Results
exp(x)	x	1	$1 + x$
exp(x)	x	2	$1 + x + x^2 / 2$
exp(x)	x	3	$1 + x + x^2 / 2 + x^3 / 6$
log(x)	x=1	3	$x - 1 - (x - 1)^2 / 2 + 0.33 (x - 1)^3$



```
| f(x)   | x   | n | Results                               |
|--------+-----+---+---------------------------------------|
| exp(x) | x   | 1 | 1 + x                                 |
| exp(x) | x   | 2 | 1 + x + x^2 / 2                       |
| exp(x) | x   | 3 | 1 + x + x^2 / 2 + x^3 / 6             |
| log(x) | x=1 | 3 | x - 1 - (x - 1)^2 / 2 + 0.33 (x - 1)^3 |
#+TBLFM: $4=taylor($1, $2, $3);n2
```

Export

Press C-c C-e.



```
Use SPC, DEL, C-n or C-p to navigate.
[C-b] Body only:     Off          [C-v] Visible only:     Off
[C-s] Export scope: Buffer        [C-f] Force publishing: Off
[C-a] Async export: Off


[c] Export to iCalendar
    [f] Current file              [a] All agenda files
    [c] Combine all agenda files

[h] Export to HTML
    [H] As HTML buffer            [h] As HTML file
    [o] As HTML file and open

[l] Export to LaTeX
    [L] As LaTeX buffer           [l] As LaTeX file
    [p] As PDF file               [o] As PDF file and open
    [B] As LaTeX buffer (Beamer)  [b] As LaTeX file (Beamer)
    [P] As PDF file (Beamer)      [O] As PDF file and open (Beamer)

[o] Export to ODT
    [o] As ODT file               [O] As ODT file and open

[t] Export to Plain Text
    [A] As ASCII buffer           [a] As ASCII file
    [L] As Latin1 buffer          [l] As Latin1 file
    [U] As UTF-8 buffer           [u] As UTF-8 file
```

Emacs is more than org-mode

- be configured as an IDE for C++ (or whatever)
- manage mailing lists
- chat on IRC
- browse web pages (eww)
- play games
- everything including making coffee

Start your journey!

Find your motivation

Dive into it when necessary

Emacs Lisp is not a must, but it is helpful.

Have fun!

Wait! . . . But where to start?

- go through two tutorials:
  1. M-x help-with-tutorial (a tutorial shipped with Emacs)
  2. C-h i h (the info-mode tutorial)
- Github | mastering-emacs-in-one-year-guide
- Github | Spacemacs-rocks
- emacs.sexy
- Planet Emacsen
- your own exploration

Thank you!

slides made with org-mode & LaTeX