Notes on Emacs and Org-mode

Ryan G

March 5, 2020

Contents

Spacemacs is slow

Current Speed up workflow

1. Put everything in memory

```
vmtouch -vt ~/.emacs.d/
vmtouch -vt ~/Notes
```

2. load all agenda files as buffers.

```
(defun open-all-org-agenda-files ()
  (interactive) (let (
    (files (org-agenda-files)))
      (mapcar (lambda (x) (find-file x)) files)))
(open-all-org-agenda-files)
```

Other tips

So if this does become a deal breaker switch to *Doom* and fix the following

- map `Spc Spc` to `Helm-M-x`
- Figure out how to fold source blocks
- Map / install Helm-Rifle to keybindings

- Often when `helm-org-rifle` stops before dispatching it is generating LaTeX fragments as I mentioned here. So don't delete the fragments and tolerate them until you can actually figure out how to fix them. (so `rm ./ltximg/*` is bad).
  - Just for a note `, T e` to preview pretty symbols is quite good.

Throw it in RAM

OK so I'm not sure if it's my imagination or not, but throwing emacs in ram seems to make it feel smoother, regardless, emacs and all my org notes are 2.5 % of my total system memory and this is like my main tool so I don't see why I wouldn't use memory I paid so much for.

I got this hint from this blog and the idea is to use vmtouch to load everything into memory, so at startup, have the following run:

```
# -v is verbose
# -t is *T*ouch into memory
# How much is currently in memory?
vmtouch ~/.emacs.d/
# Put it in memory
vmtouch -vt ~/.emacs.d/
vmtouch -vt ~/Notes/Org

# Now how much is in memory??
vmtouch ~/.emacs.d
```

make sure that you throw it into an

Potential Fixes

1. Use new frames when opening a document simultaneously
2. Use a standalone frame for a given document
   (a) Especially long documents
3. Just Restart it
   (a) `killall emacs`, `systemctl --user stop emacs`
4. Delete the directory and reinstall it.
   (a) This sometimes makes a whopping! difference for me
      i. But I have to do it far to regularly on Spacemacs and it makes me want to switch to doom for good.

Code Folding

in vim to fold code you use the following syntax:

```
## vim:fdm=expr:fdl=0
## vim:fde=getline(v\:lnum)=~'^##'?'>'.(matchend(getline(v\:lnum),'##*'
    )-2)\:'='
```

DONE Making a Schedule in Emacs

I think just make a couple todo items, add them to the agenda and give them times? should I learn more formally or work on it as I go along? Maybe both, play with it for a few days then read the manual on spacemacs and org-mode.org

DONE Seperate out LaTeX template components

Actually, no, just set up LaTeX folding in *Emacs* and make nice pretty templatex, because, when you need to use a template in `org-mode` it's nice being able to call a single package and a package can't call relative directories.

Well actually I'm supposed to be able to set up relative directories with STY files.

DONE How do I have text flow across windows?

The appropriate width of text is taken to be 50 characters [**zotero-53**], however some research has shown limited difference between 35-95 [**Cha**].

There is some evidence to suggest a two column full justified layout may be better for reading digital material [**Mun**] and research shows that 8 cm (roughly two columns) text width is more readable in print [**Tin1929**], it also stands to reason that this would give me the opportunity to use smaller margins, perhaps this is something I should put in my default LaTeX templates?

The preferred font size in digital media tends to be taken as quite large at 16 points [**Che1996**],

And there is a general, albeit slight tendency to prefer Sans Serif *Verdana* font in digital print [**Ali2013**, **Hoj2014**]

Regardless, for code, the *Tidyverse* style guide sets a limit of 80 characters [**Wic**] and for that reason I need to set up columns that flow in emacs, like `follow-mode`, but, without the horrendous lag.

I figured this out, it wasn't the =follow-mode per se, it was:

1. use Spacemacs base
2. Clear out Crap from `~/.emacs.d/init.el`

   [**worldParachuteDeviceFair1939**]

DONE How do I use LaTeX snippets Snippets

- Is there anything built into spacemacs?
  - there is `yasnippets` but it's going to be a bit of work to implement it
- Is it close enough to what I've been using?
  - it's not but it sounds like the tex mode might be
- should I just jump back to vim when necessary?
  - for now yes, but maybe long-term something could be implemented.
  - for live tex preview you can always open a new vim-buffer with an md extension and then use `<leader>-lv` to watch the TeX form as md while you write it and then paste that back into the `.org` file.

Ok so I could not get company snippets or electric/auto snippets to work what did work though was using `helm-yas` (`SPC i s`), that's definitely the way to go

Official Snippets

You're probably better off using a package of official snippets to:

1. save time
2. stay on a sound platform

DONE Example LaTeX

previewing latex

Right, let's take a math problem, in this case I'll use something from *Mathematical Modelling*:

According to the manual, in order to change the defaults of the image preview, use `M-x customize RET org-format-latex-options`:

- set foreground to 'auto'
  - this is single `'`
- set background to "Transparent"
  - this is double `"` because it's expecting a `stringp` argument
- in order to reset the images use rm `./ltximg/*`
  - be careful, I deleted everything by putting `*` in the wrong place, don't do that.
  - Generally this a bad idea, don't do this it slows everything down, instead, make a transperant preview with like a vivid blue/green/red color to it, that way it will be visible on both.

there appears to be a bug with this however and the foreground colour can only be black, which can't be seen on a black background.

Instead you need to change the variable `org-preview-latex-default-process`

```
;  Either this
  (setq org-preview-latex-default-process `dvisvgm)
; Or this but not not Both?
  (setq org-preview-latex-default-process `dvipng)
```

The idea is now that the option in `org-format-latex-options` will be respected for the foreground colour

and then the settings should be such that the foreground and bacground are both set to `default`.

$$math4x$$

$$\frac{1}{x^2} \times \left[2x \cdot \frac{\mathrm{d}\, y}{\mathrm{d}\, x} + x^2 \frac{\mathrm{d}}{\mathrm{d}\, x}\left(\frac{\mathrm{d}\, y}{\mathrm{d}\, x}\right)\right]$$

Thisanswer uses xcolor and dvipng to get around it but the svg solution works to.

Now this is also a problem with AuCTeX which doesn't have the luxury of using `dvisvgm` and only works with `dvipng`, so this needs to be sorted out.

by using `customize-variable` to inspect `preview-pdf-adjust-color-method` (which I was linked to by `preview-transparent-color` because they are both in the `preview-apperance` group in theAuctex manual) it was suggested that I need to be using the next version of ghostscript which jumped from 9.27 to 9.50, Ubuntu is still on 9.50, so I've installed, from source, 9.50, the default in ubunu is 9.26 located at `/usr/bin/ghostscript`, the newer one, that I compiled from source is located at `/usr/loca/bin/gs` so I moved the old one to `ghostscript.bak` and symlinked the new one in place.

After that AucTeX started working, but I couldn't get dvipng to work for org mode so It's necessary to use dvisvgm for org-mode, as for tikz, you need to be aware that the colours are user defined, so you might want to define the colours like so: `\begin{tikzpicture}[domain = 0:10, scale = (2/3), draw = white, text = white]`

1. imagemagick failure I was getting the error:

   ```
   File mode specification error: (error File "/tmp/orgtex9SIJrf.png" wasn't produced.  Pleas
   ```

   in order to fix it I needed to:

   (a) `sudo vim /etc/ImageMagick-6/policy.xml`

   (b) Change the line `<policy domain"coder" rights="none" pattern="PDF" />` to: `<policy domain="coder" rights="read|write" pattern="PDF" />`

   I also found that after fixing this I no longer needed to use `dvisvgm` and so I just commented it out of the init file.

   (a) LaTeX Header is important!LaTeX:Org So I didn't understand at first how this all worked but now I do, the process is pretty basic really, when you put your cursor on `$ math $` and hit = C-c C-x C-l= :

i. The contents captured is taken to a seperate buffer
ii. that buffer is exported in the expected way to LaTeX
    A. This means that the typical `\usepcackage{}` crap is put in there, but also your header as well.
    B. so if your LaTeX style doesn't exist or points to a bad reference even inline images of math won't render.
iii. The LaTeX `dvi` / `pdf` is then converted to a `png` or `svg` using `imagemagick` or `dvisvgm`

so if you're having trouble, first comment out the LaTeX header and then retry because that could very well be the issue.

If that does fix it, it's probably a `references.bib` that's become erroneous, that's what it always is.

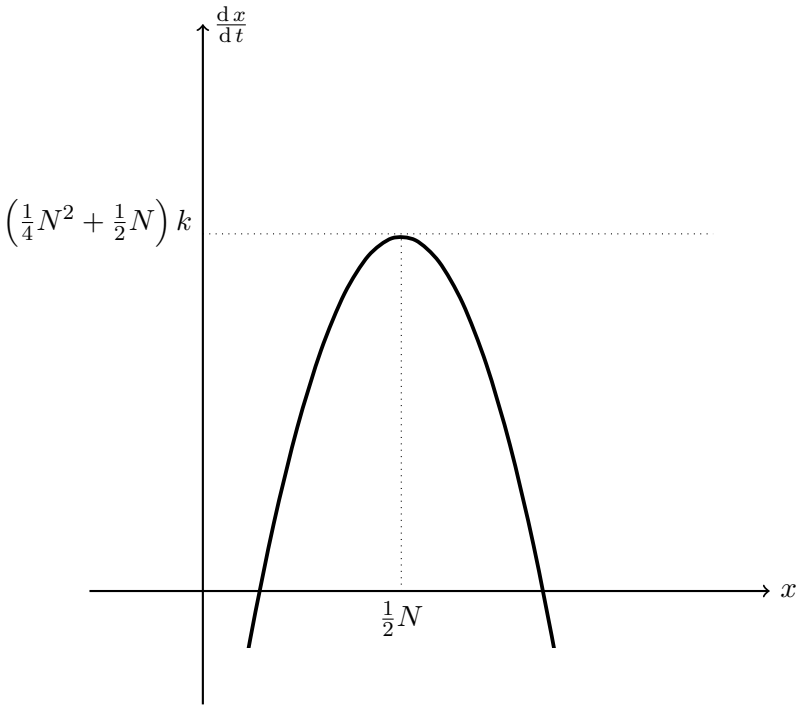
Also important is that you use a `BibTeX` not a `BibLaTeX` export from *Zotero*, there is no support for `@online` in `org-ref` so make sure `BibTeX` specifies them as `@misc`

DONE Using the snippets

Take the equation and open it in `org-edit-special` mode with `C-c ' / , '` :

$$\frac{1}{x^2} \times \left[2x \cdot \frac{\mathrm{d}\, y}{\mathrm{d}\, x} + x^2 \frac{\mathrm{d}}{\mathrm{d}\, x}\left(\frac{\mathrm{d}\, y}{\mathrm{d}\, x}\right)\right] \tag{1}$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2}$$



Then just hit `SPC i s` to engage `Spacemacs/helm-yas`.

It's honestly easier to just hit `C-c v` and then use the vim live preview, it will still work despite the document being an `org-mode` file.

honestly, if you need fast vim with a preview, and/or tikz, this might be the simplest way:

1. open a new destkop in i3/cinnamon
2. open a terminal and create a new directory
3. Either use vim to create a `temp.tex` and have the default template be pulled in (or use the older `texnote` command)
4. edit in that window leveraging the almost live preview offered by vimtex
5. use zf to fold and `<Spc> y` to copy relevant things to the clipboard.
6. paste it back into emacs
    - if you get better with i3 you could probably have a really elegant way to do this on one desktop, may `mod a` a couple times and `mod enter` works pretty well tbh.

I don't see any much need to re invent the wheel with snippets when vim works well and I can't sync them between so I'm just going to leave it.

DONE Fix `outline-minor-mode` hook for LaTeX

Preview LaTeX

LaTeX may be previewed from the `LaTeX-mode` by using `SPC m v` or `, m v`, in order to change the default to `zathura` change the variable `TeX-view-program-selection`

DONE How to troubleshoot lag by looking at cpu times?

Spc m is undefined Workflow

Look this happens all the time, I don't think it's really an issue though, just make sure to execute `M-x toggle-org-custom-inline-style` and then just export dispatch with `C-c C-e`

DONE Emacs for notes

<2019-11-13 Wed> You can add a todo tag by calling `org-insert-todo-heading` which is mapped to `M-Shift-RET`, in *SpaceMacs* it appears to be `SPC m :`

Consider adding extra states to the for todo [1] something like:

```
#+TODO: TODO IN-PROGRESS WAITING DONE
```

[1] I got this from theO'Toole Tutorial

Using Helm

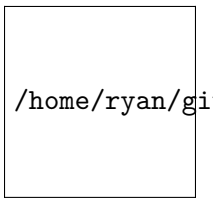
You should read through the Helm Tutorial, it's really helpful [2]

DONE Images diagrams

Images may be inserted as links

Look at the following example:

The documentation implies that `[[ ]]` are required but it seems that the export works fine without that and the inline preview is fine without it as well:



This may be inline previewed with `C-c C-x C-v`

Resizing Images

Images may be resized by first specifying the following in emacs:

```
(setq org-image-actual-width nil)
(setq org-latex-image-default-width "")
```

and then using the following syntax:

```

```

DONE References bibliography

DONE BibTeX mode

this is a major mode to edit `.bib` BibTeX files, to use this you need to enable the `(bibtex)` layer in the `~/.spacemacs` file and then set up a basic config. [**Ben2019**]

This works really well because you can tie the reference to a PDF on the system (Dropbox or Git should make laptop in sync) and then when you press enter you can open the pdf, search the citation, email it etc. all from in Emacs.

Inserting references is really good to because you can just jump straight to the `.bib` file and/or google scholar.

[2] A Package in a league of its own: <code>Helm</code>

basically the only way I've found to change the default bib file is to run the following inside the file:

For the moment I've decided not to do that, just to trial having only a single bib file, because it is easy to filter a bib file using the BibTeX major mode in Emacs, just use `C-c a` and then you can filter it by fields etc, it probably makes more sense just to have one single BibTeX and if you get really lost use Zotero or Jabref manage it with tags, tho whole idea of a `.bib` file is it's an index so it might not be worth parting it out (or you could comment sections out if you really wanted to, it's better than hunting for files).

To add a citation in an org file use `SPC m i c` from there you can look something up with google or whatever or go to the bib file and search for the keyword from there. From the bib file you can use ISBN or DOI to autogenerate an entry also there is the `org-ref-url-html-to-bibtex` function, which performs well (even getting the year of publication when *Zotero* didn't) but *Zotero* links back to a cached version of the HTML which is really nice.

As an example I have cited the NMR paper I was supposed to understand [**Blo1957**]

It's extremely important is that you remember to NEVER use the ~ character in LaTeX, relative and absolute paths are both supported but that character is not supported.

What's extremely important is that you remember to NEVER use the ~ character in LaTeX, relative and absolute paths are both supported but that character is not supported.

Also you have to remember that listings will fail if you use use `elisp`, you MUST use `lisp`, be mindful also that `elisp` will configure emacs but `lisp` will not.

You will also have to specify, at the very end of the org file, a bibliography using the following syntax, (it must be at the end, which is frustrating):

Now also really important, all of that code MUST go into the `~/.spacemacs` in the `user-config` section, that way it will be loaded earlier, atleast I think it does because it started working for PDF after I did that (but still not HTML, and the internal link following broke too.)

Now as for comments in a .bib file, these are a nuisance because many interpreters (including those used by texstudio and texvim will include things outside the braces.

1. Managing Citations The BibTeX major mode is so good for Emacs I almost don't want to use anything else, but, the Zotero better BibTeX integration is really fucking amazing, and does everything that the emacs integration does, it only syncs one way though so you will have to leave emacs which is a pain if your on i3 but nice generally.

   It caches websites which is quite helpful but it might somewhat break helm integration.

   Because you can't use comments in a .bib file though, it might be well worth looking at using zotero, there's no fear about it disappearing either because it's free and open source.

DONE org-ref

The problem is exporting doesn't work, BibTeX is just a major mode that only manages the creation of citations in the file and the behaviour of Emacs regarding links to entries in BibTeX files, in order to export and leverage citations you need to use the org-ref package.

you definitely want to use the help `org-ref-help`, don't bother with the manual, read the help instead.

That link will show you what to add to the config file as well as how to configure the `org-latex-pdf-process` to use bibtex, basically you need to do the following:

1. Decide how you are going to manage references

   - Emacs and the `BibTeX` layer are pretty good and will more or less just work, the only issue is you will need an add on in Firefox to generate bibtex.

- Zotero is phenomonal and can be tied to autopush to a BibTeX using the *Better BibTeX* extension, it supports caching websites as well which could be handy. Its free and open source so theres no reason not to use it other than a desire to become more familiar with `.bib` files.

2. Set a default location for:
   - the main `.bib`
   - the `notes.org` corresponding to that `.bib`
   - the folder containing PDFs linked to that `.bib`
     - The PDFs MUST match the name of the key.
     - I think you could probably just link to zotero, or, zotero might do everything on an export?
3. set the main location for the `reftex` bib
4. At the top of the `.org` document set the corresponding location for the `.bib` used by that file:

```
#+latex_header:
 ↪ \addbibresource{/home/ryan/Dropbox/Studies/Papers/references.bib}
```

1. Change the LaTeX build option to tolerate broken links.
2. Change the LaTeX build option to build the BibTeX
   (a) You will need to remember the `-shel-escape` flag for the `minted` package.
   (b) This method automates the process so I *think* it should work for XeLaTeX and LuaLaTeX as well, but i'm not sure.
3. Pass the bibliography info to LaTeX by placing the following lines at the end of the `.org` file, all the `.bib` references should match, although I remember reading that you may be able to specify multiple with commas, I wouldn't bother.

```
<<bibliographystyle link>>
bibliographystyle:unsrt

<<bibliography link>>
bibliography:/home/ryan/Dropbox/Studies/Papers/references.bib
```

So all-up you would need to add the following to you `.emacs.d/init.el`, however I put all of the configuration for BibTeX stuff in the `(defun dotspacemacs/user-config ()` section of the `~.spacemacs`, because intuition suggested to me that the loading order mattered for this:

```
;; ;2. Set Default Location; see org-ref for use of these variables
(setq org-ref-bibliography-notes
      "/home/ryan/Dropbox/Studies/Papers/notes.org"
      org-ref-default-bibliography
       '("/home/ryan/Dropbox/Studies/Papers/references.bib")
      org-ref-pdf-directory "/home/ryan/Dropbox/Studies/Papers/")

; 3. Set the Reftex Location
(setq reftex-default-bibliography
      '("~/Dropbox/bibliography/references.bib"))


; 5. Tolerate broken links
(require 'org-ref)
(setq org-export-with-broken-links t)
; This finding broken links might be slow in big files, feel fre to set
    as nil
(setq org-ref-show-broken-links t)

; 6. Change the build Process
; Build LaTeX with BibTeX option
(setq org-latex-pdf-process (list "latexmk -shell-escape -bibtex -f
    -pdf %f"))

(setq org-latex-prefer-user-labels t)

;;;;; Helpful packages
(require 'org-id)
(require 'org-ref-wos)
(require 'org-ref-scopus)
(require 'org-ref-pubmed)
```

Be careful though, don't add everything listed in their GitHub installation instructions, it might break the HELM stuff.

To get the export depends on whether or not you are using bibtex or biblatex, bibtex is slightly simpler, but once it's all working switching to BibLaTeX shouldn't be hard. Once this is set up be mindful that the LaTeX export will only include references to exports in buffers if you now tie in the appropriate headers and footers etc. That's a little annoying, however I've just put the BibLaTeX stuff into my template because I've decided just to use that from now on, BibTeX is only necessary to submitting to journals. [**zotero-45**]

1. BibTeX Those previous instructions set everything up for BibTeX, on order for the citations to be compiled you **must** include a bilbiography link as in those instructions, like, don't change them.

   (a) **DONE** What is a bibliography link the `org-ref-help` instructions instruct to use a `bibliography link` and a `bibliographystyle link` these lines at the end of an `org-file` that tell `org-ref` what the bibliography files are and where to put them in LaTeX:

```
<<bibliographystyle link>>
bibliographystyle:unsrt

<<bibliography link>>
bibliography:/home/ryan/Dropbox/Studies/Papers/references.bib
```

2. BibLaTeX The thing is BibTeX is old and doesn't work well with URL's, if you want to start referencing everything properly and incorporate referencing in to your daily workflow, you're going to have to use BibLaTeX.

   before doing anything, because LaTeX is so extremely old you MUST ensure the following:

   - all old `.aux` files are gone
   - all old `.bbl` files are gone
   - section names must not contain
     - footnotes
     - links
     - math (this should work but often breaks with `href`)
   - If you use languages like `markdown`, `emacs-lisp` etc. you switch the `minted` package, because otherwise listing will stop the whole thing.
   - the `hyperref` package is loaded before everything else.

   when using `biblatex` follow on from the previous BibTeX set up and modify the following:

   (a) Open *TeXStudio*
      i. Change from `BibTeX` to `biber`
      ii. Make sure to add the `-shell-escape` flag if you're using minted
      iii. open the `.tex` file, `.bib` file and any `.sty` files etc. so that they can be troubleshooted.

   (b) Just leave the `.bib` file as is, you could modify it to include things like `@online` rather than `@misc` but it's going to break exports later on so just leave it. But If you wanted to, you would put the following in your config, but just leave it.

```
;Use BibLaTeX not BibTeX
(setq bibtex-dialect 'biblatex)
```

   (a) Specify the style and bibliography file at the top of the org file like this.

```
#+latex_header: \usepackage[citestyle=authoryear-icomp,
↪ bibstyle=authoryear,hyperref=true,backref=true,
↪ maxcitenames=3,url=true,backend=biber,natbib=true]{biblatex}
#+latex_header: \addbibresource{/home/ryan/Dropbox/Studies/Papers/r⌋
↪ eferences.bib}
#+latex_header: \AtEndDocument{\printbibliography}
```

- It's probably better to put all of this in a style sheet, then just call that style sheet as a LaTeX header:
  - This has the advantage of meaning you will get a bibliography in an `indirect-buffer` just like you do in HTML, that's what i've done in this document actually.
  - You're going to want to also use `hyperref` with citatations, so you can load and set up `hyperref`, which must be done first thing, and load the bib stuff in one fell swoop.

(a) Remove the `BibTeX` links at the end

```
\* References

# ####### Delete these!!!!!
+<<bibliographystyle link>>
bibliographystyle:unsrt

<<bibliography link>>
bibliography:/home/ryan/Dropbox/Studies/Papers/references.bib
# ####### Delete these!!!!!
```

- It seems the `* References \n` heading is necessary, `org-ref` needs that to figure out whether or not to include anything beneathe it in the LaTeX documents. Despite this, LaTeX will always print a `section{References}` when it sees `printbibliography`
  - For this reason it makes more sense to include the `printbibliography` in the `AtEndDocument` macro in the header.
- HTML will continue to use `BibTeX2HTML`, which is just `BibTeX`, so it won't work at all with BibLaTeX, but the implementation with BibTeX is awful anyway and so instead `citeproc` will be used to fix it.
  - One of the problems is that I really need to be using `biblatex` for half descent references, but, I also need those references in HTML.

4. HTML References When using `org-ref`:

- BibTeX will give you a half-ass HTML and a half-ass LaTeX
  - For BibTeX for example requires putting URL's in the `notes` field
- BibLaTeX will give you a decent PDF citation style (and I think supports harvard) but no HTML whatsoever at all.

So for HTML we need to set up `citeproc`, `org-ref` includes a function `orcp-citeproc` that can be hooked in before the export function, but it's not very good and I couldn't figure out how to specify custom styles or hyperlinks, the newer package `citeproc-org` however supports hyperlinks and custom `.cls`, so it's worth moving to this one even if it is a bit incomplete.

This method does not support the use of reference types like `@online`, and so for this reason a BibLaTeX file is not supported and a BibTeX file must be used. This method does however support the `url` field, so, just have the *Better BibTeX* add-on link to a file where it exports with the `url` field (it is also possible to export URL's to the `notes` field as a fallback). BibLaTex and

LaTex will actually use the `url` fields, the only thing that will change is that the type will be set as `@misc`, when you're doing an actual paper and need referencing to match some 100% asanine definition, just re-export the zotero DB as BibLaTeX and change the LaTeX preamble to reflect the new DB location.

- This might cause a few other styles to fail, the `Nature.csl` style I'm using seems to just work so I'm going to keep using that, but, I modified the `XML` to use `[]` rather than `()` beccause they're less ambiguous.
- It started playing up so what I did was I replaced the `chicago` style file and I moved the `url` field to a `notes` field, I just don't have the time for this to not *just work* so it's better to just leave it like that.

In order to install and set up `citeproc`:

(a) install `citeproc` (use this method otherwise spacemacs removes any installations as as orphans)
   - add `citeproc` and `citeproc-org` to the `additional-packages` of `~/.spacemacs`
   - use `spacemacs/paradox-list-packages`
     - don't use `M-x package-install RET citeproc`

(b) install `citeproc-org` by using `package-install-file` and finding the file downloaded fromthe GitHub repo

(c) you need to put the `locales` and `styles` from the repo into the directory, I think it is `~/.emacs.d/elpa/citeproc-org-0.2.2`, so just clone the repo into that directory basically.
   i. This isn't specified on the github, but otherwise it fails looking for a file in that location so clearly it needs those files.

(d) Now that it's installed you need to set up the hook, this can be done by defining and running the following function:
   - the `orcp-citeproc` is one created by Kitchin for `org-ref` it's incompatible and the `citeproc-org` one is much better, just make sure the the `ocrp` hook is removed.

(a) The style is by default Chicago, you need to download another one from *Zotero* and specify at the top with `+#CSL_STYLE`, This is a good default.
   i. This is where problems can occur though because some citation styles don't work with certain fields, like `url` and owing to this citeproc will fail, the Nature one appears to just work so rename the Nature one that I linked above to the Chicago one which acts as a default (i.e. the one used on indirect buffers)
      A. If you still have problems, just export the URL's from *Zotero* as a `notes` field rather than a `URL` field, which is what I've done, but the *Nature* style won't then show `notes` but it will show `URL's` so that's confusing.
   ii. It is necessary to use a BibTeX file because things like `@online` will prevent this from working.

(b) Done, now you should have BibLaTeX and decent HTML referencing, `org-citeproc` overtakes citation rendering for non-LaTeX `org-mode` export backends [3],

[3]

```
(defun citeproc-org-setup ()
  "Add citeproc-org rendering to the `org-export-before-parsing-hook'
  ↪ hook."
  (interactive)
  (remove-hook 'org-export-before-parsing-hook 'orcp-citeproc)
  (add-hook 'org-export-before-parsing-hook
  ↪ 'citeproc-org-render-references))
(defun citeproc-off ()
  "remove citeproc-org rendering from the
  ↪ `org-export-before-parsing-hook' hook."
  (interactive)
  (remove-hook 'org-export-before-parsing-hook
  ↪ 'citeproc-org-render-references))

(citeproc-org-setup)
```

Listing 1: Source Code to define function to enable hook, run this function to enable citeproc inside a document

(c) Now, you need to be a little mindful of the hook though, all the exports with the exception of LaTeX will now all be formatted, no more referencing to the BibTeX, this may be acceptable, it may not be, you may need to disable hook with `citeproc-off` if you don't want it and fall back to a pure `org-ref` in order to allow `org-ref` to export the BibTeX to other formats rather than that formatted by `citeproc`.

- `citeproc-org` does not render citations for export backends that are on the list `citeproc-org-ignore-`
  Citation rendering for these backends is handed over to the active default rendering mechanism (`org-ref`, for example, uses BibTeX or BibLaTeX for LaTeX and `beamer` backends).

The reason for all of this, really, is that I need to practice with BibTeX and referencing so as long as I put anything in it doesn't matter [**Gis1990**] , ,it has been shown that study habits, skills and attitude towards learning are as indicative of academic success as past performance [**Cre2008**]

5. Reverting to `org-ref` for export If you need to revert to `org-ref` for the citation manager:
   - Put the bibliography link at the end of the org document were deleted at example **??**
   - Remove the hook by using `citeproc-off` as defined in source code **??**
     - you can't toggle a hook so
6. Using Zotero vs Using BibTeX layer As nice as the BibTeX layer is, Zotero will automatically download PDF's as well as the actual html, it also supports taking notes in the actual application, it's going to be too difficult to get BibTeX to download them through the school proxy for me and I am going to need to be able to cite papers.

   (a) Considerations of BibTeX vs BibLaTeX many journals don't support BibLaTex, using zotero might make it easier to jump between in a pinch, however, you could always import into zotero the limited options of BibTeX/BibLaTeX vs *Zotero* may impede it.

Internal references

I just took this straight from the help `org-ref-help`

1. ref links

   A ref link refers to a label of some sort. For example, you can refer to a table name, e.g. Table **??**. You have to provide the context before the ref link, e.g. Table, Figure, Equation, Section, and so on.

   Table 1: A simple table.

x
1
2

   Or you can refer to an org-mode label as in Table **??**.

   Note: You may need to set org-latex-prefer-user-labels to t if you refer to times by their "name" for the export to use the name you create.

   ```
   (setq org-latex-prefer-user-labels t)
   ```

   Table 2: Another simple table.

Command	Result
`false; echo "yes"`	prints yes
`true; echo "yes"`	prints yes
`false && echo "yes"`	does nothing
`true && echo "yes"`	prints yes
`firefox & disown`	runs firefox and then disowns

   To help you insert ref links, use the "Org -> org-ref -> Insert ref" menu, or run the command `org-ref-helm-insert-ref-link`. There is no default key-binding for this.

   ref links are functional. If you put the cursor on a ref link, you should see a message in the minibuffer with some context of the corresponding label. If you click on the ref link, the cursor will jump to the label.

   A brief note about references to a section. You can make a ref link to a $\text{CUSTOM}_{\text{ID}}$. Section **??** has a label link in the headline. That works, but is not too pretty. Section **??** uses the $\text{CUSTOM}_{\text{ID}}$ property. For this to work, you should set `org-latex-prefer-user-labels` to t.

   Also note that "#+tblname:" and "#+label:" are deprecated in org-mode now, and "#+name:" is preferred.

2. Miscellaneous ref links

   org-ref also provides these links:

   pageref The page a label is on

   nameref The name of a section a label is in

eqref Puts the equation number in parentheses

autoref A command from hyperref that automatically prefixes the reference number.

cref & Cref cleveref – Intelligent cross-referencing (crefrange is not supported)

Note for eqref, you must use a LaTeX label like this:

$$e^x = 4 \tag{3}$$

Then you can refer to Eq. (**??**) in your documents.

Autoref works like this: **??**, **??**.

You can specify the default ref link type in 'org-ref-default-ref-type'.

3. Inserting Links You can store a link from emacs into org using `SPC a o`, this link can be later pasted with `C-c l`, you will want to adjust the variable `org-link-file-path-type`, it allows relative paths etc. to be toggled.

   If the link contains `docview::` it will be exported as absolute, so always take the link to things like PDF files from *Treemacs*, never from the docview buffer, always check the link for `file:` which is what you need.

Different bib file for a single document

If you want to change the reference file just for just one project, you can tell BibteX you are using that file somehow and you can just change all the references in the file, I wouldn't bother though, I'd just use one massive file because BibLaTeX will only include what's cited and you can use programs to strip the `.bib` file apart when given a LaTeX file with corresponding entries. (or you could use a vim macro to be honest).

DONE How do attach external files that I can follow? Attachments

Does the filetype matter? like markdown pdf whatever? This Link to the manual suggests that any file can be linked to be using `C-u C-c C-l`PDF to this doc, pretend it's a tutorial sheet or a learning guide or something

You may also want to consider using the insert menu `, i a` / `<SPC> i a` to reach the attach menu `org-attach` / `, i a a` `<SPC> i a a` which might do a lot of the work for you:

Is there a way to include links to locations of external files?

Interlinking

There is a tonne of information in the manual on how to do this, but The easiest way is to use: (`SPC a o l` followed by `C-c l` is your friend) fs

- Org-brain

- Org-Wiki
  - when org is exported to HTML, the links are changed to `.html`, so it will correspond to HTML files in that directory, an auto export hook could solve that.
- Hyperbole

One could also have auto HTML export as well by using this lisp code:

```
# Local variables:
# after-save-hook: org-html-export-to-html
# end:
```

Mathematics LaTeX

Inline Preview

When using `C-c C-x C-l` to preview LaTeX fragments, make sure you are using `dvi2png` because it is way faster, avoid using `imagemagick` because it is slower.

DONE Mathml Mathml

The doc string for `org-export-with-latex` describes how to select mathjax, verbatim output, or not to export LaTeX at all, although it goes further to say:

> If you prefer, you can also request that LaTeX fragments are processed into small images that will be inserted into the browser page.

But no value is actually listed for requesting that.
The only way I've obtained the desired result is with:

- `#+OPTIONS: tex:dvipng`
- `#+OPTIONS: tex:dvisvg`

If this is needed globally set the variable specific to the `html` backend (or preferably the email backend):

```
(setq-default org-html-with-latex `dvipng)
```

1. Equation References

   org-ref also provides these links:

   pageref The page a label is on

   nameref The name of a section a label is in

   eqref Puts the equation number in parentheses

   autoref A command from hyperref that automatically prefixes the reference number.

cref & Cref cleveref – Intelligent cross-referencing (crefrange is not supported)

Note for eqref, you must use a LaTeX label like this:

$$e^x = 4 \tag{4}$$

:foreground default :background default :scale 2.0 :html-foreground Black :html-background Tr

Then you can refer to Eq. (**??**) in your documents.

Autoref works like this: **??**, **??**.

You can specify the default ref link type in 'org-ref-default-ref-type'.

LaTeX Export

It is extremely important not to have filenames with whitespace characters, this WILL break LaTeX exports when using the minted package owing (I believe :shrug:) to the `-shell-escape` flag.

DONE Tag Filtering tagging

General Information

- According to the documentation on Tag Inheritance, Tags respect the heirachy of the document.
- you can also put (`setq org-complete-tags-always-offer-all-agenda-tags t`) in the `~/.emacs.d/init.e` in order to have autocompletion from all other documents in the agenda.

1. Global Tags In order for tags to appear in the agenda buffer they must be added to the `org-agenda-files` variable, Files can be added /removed from this variable with `C-c [` / `C-c ]` respectively.

   use `C-c a` / `<SPC> m a` to get up the agenda view, from there you can open an *agenda-buffer*:

   - `m` will take you to tag matching
     - `M` will take you to tag matching for items marked `todo`
   - `t` will take you to all the todo lists
     - `T` will let you choose `WAITING=/=done` etc.

   If I do agenda search with `C-c a` then I can further filter with `org-agenda-filter-tags` tied do \, but the tags listed aren't pruned which is a problem, can I filter based on the tags based on the top first or between?

   After the agenda buffer has been generated you can use \ to filter by tag or use `<ALT>-<SPC>` to open the transient state, which will offer many different ways to filter the tags.

   You can't filter which tags popup when you try and filter by tags, so the best method, for now, is to use \ tag by tag and read-off whats in the list one-by one, using nested tags and sensible names will help.

   In order to have all the global tags listed, you need to the following in the `~/.emacs.d/.init.el`:

```
(setq org-complete-tags-always-offer-all-agenda-tags t)
```

(a) Progressive Filtering I haven't found anyway to do do this, but `org-global-tags-completion-table` might be a start.

2. LocalTags Local tags are managed by *Sparse Trees* which collapses and flattens the tree but for matching tags, certainly preferable for a local document.

   using `,` `/` allows me to create *Sparse Trees*, I can choose to do that either by todo status or by tag, although this broke in these notes so it may not be reliable?

   After creating the sparse tree I don't believe further filtering is possible, however you can use *Boolean Logic* when creating the tree in order to have it even sparser.

   it appears that if I press `<SPC> m a m` (for mode-agenda-match) I can search for all tags, it also implies that tags apply per headline? A local sparse tree can be generated by using `c-c / m` or `C-c \` First you need to list the org-file in the variable `org-agenda-files`, this could be customised manually but you usually just do `C-c [`.

   - if you press `<SPC> m b` it will take you to a buffer only with what is beneath that headline, this is amazing, just like *Beorg* , really good!
     - to kill this buffer: `<SPC> m b / C-x k`
     - it's also possible to jump straight to the buffer list with `C-x C-b` and:
       (a) mark buffers for deletion with `d d=/=C-k`
       (b) eXecute the deletion with `x`

3. Listing Empty Headlines use the special property TAGS like so:

   ```
   TAGS=""
   ```

4. Tag Heirachy If you list tags in the preamble or `org-tag-alist` like this:

   ```
   #+TAGS: [ Control : Context Task ]
   #+TAGS: [ Persp : Vision Goal AOF Project ]
   ```

   you will get a scheme like this:

   - 'GTD'
     - 'Persp'
       * 'Vision'
       * 'Goal'
       * 'AOF'
       * 'Project'
     - 'Control'
       * 'Context'
       * 'Task'

So if you search for 'GTD' you will get back everything beneath that (e.g. 'persp' and 'Goal'), if you search for 'Control' you will only get back matches for 'context' and 'task')

This is better than using tag inheritance because something tagged as 'PCA' may not necessarily be underneath a heading like "Unsupervised" Learning.

It's probably better to set these up in the `org-tag-alist` variable in the config but I can't get it to work, so instead list them in the preamble and leverage the fact that all files in the agenda will be listed as valid tags.

5. **DONE** Capture Templates Say I want to take a math note, can I just use <SPC a o c> and open the right template? There is some info in the Manual and on This Site.

TimeStapms

Type	Description
Deadline	When a task is due
Scheduled	When you intend to start working on that task
Timestamp	a specification of a date or a range ofdates.

Keyboard Shortcuts

Shortcut	Description
C-c / m , / or C-c \ or <SPC> m /	Create a sparse tree with all headlines matching a tags search.
C-c [	Adds the current file to the agenda variable so it is included in agenda searcl
C-'	cycle through agenda file list
<SPC> a m	Create a global list oftag matches from all agenda files (files must be listed i
<SPC> a t	Create a global to do list
\	Only works in an agenda bugger, filters that buffr by the given tag.
(* or %) then B	Bulk Change matches in agenda view
C-c C-q	et tag for current heading
'C-u C-c C-q' / <SPC> u C-c C-q	realign tag in all heading
C-c C-o	globally (agenda) match tag at curor
C-c .	insert SCHEDULE timestamp via prompt
C-c C-d	insert DEADLINE timestamp
C-c C-	insert CHEDULED timestamp
C-c / d	create pare tree with all deadline due
C-c C-x C-i	start clock on current item
C-c C-x C-o/x	stop/cancel clock on current item
C-c C-x C-d	display total subtree times
C-c C-c	remove displayed times
C-c C-x C-r	insert/update table with clock report

there are way more more shortcuts inside this drawer [4]

```
================================================================================
Org-Mode Reference Card (for version 7.8.11)
================================================================================



================================================================================
Getting Started
================================================================================
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
restore property-dependent startup visibility     C-u C-u TAB
show the whole file, including drawers            C-u C-u C-u TAB
reveal context around point                       C-c C-r

================================================================================
Motion
================================================================================

next/previous heading                             C-c C-n/p
next/previous heading, same level                 C-c C-f/b
backward to higher level heading                  C-c C-u
jump to another place in document                 C-c C-j
previous/next plain list item                     S-UP/DOWN\notetwo

================================================================================
Structure Editing
================================================================================

insert new heading/item at current level          M-RET
insert new heading after subtree                  C-RET
insert new TODO entry/checkbox item               M-S-RET
insert TODO entry/ckbx after subtree              C-S-RET
turn (head)line into item, cycle item type        C-c -
turn item/line into headline                      C-c *
promote/demote heading                            M-LEFT/RIGHT
promote/demote current subtree                    M-S-LEFT/RIGHT
```

[4] Org-Mode Reference Card

```
move subtree/list item up/down                    M-S-UP/DOWN
sort subtree/region/plain-list                    C-c ^
clone a subtree                                   C-c C-x c
copy visible text                                 C-c C-x v
kill/copy subtree                                 C-c C-x C-w/M-w
yank subtree                                      C-c C-x C-y or C-y
narrow buffer to subtree / widen                  C-x n s/w

================================================================================
Capture - Refile - Archiving
================================================================================
capture a new item (C-u C-u = goto last)          C-c c \noteone
refile subtree (C-u C-u = goto last)              C-c C-w
archive subtree using the default command         C-c C-x C-a
move subtree to archive file                      C-c C-x C-s
toggle ARCHIVE tag / to ARCHIVE sibling           C-c C-x a/A
force cycling of an ARCHIVEd tree                 C-TAB

================================================================================
Filtering and Sparse Trees
================================================================================

construct a sparse tree by various criteria       C-c /
view TODO's in sparse tree                        C-c / t/T
global TODO list in agenda mode                   C-c a t \noteone
time sorted view of current org file              C-c a L

================================================================================
Tables
================================================================================

--------------------------------------------------------------------------------
Creating a table
--------------------------------------------------------------------------------

just start typing, e.g.                           |Name|Phone|Age RET |- TAB
convert region to table                           C-c |
... separator at least 3 spaces                   C-3 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
move to beginning/end of field                    M-a/e

-------------------------------------------------------------------------------
Row and column editing
-------------------------------------------------------------------------------

move the current column left                      M-LEFT/RIGHT
kill the current column                           M-S-LEFT
insert new column to left of cursor position      M-S-RIGHT

move the current row up/down                      M-UP/DOWN
kill the current row or horizontal line           M-S-UP
insert new row above the current row              M-S-DOWN
insert hline below (C-u : above) current row      C-c -
insert hline and move to line below it            C-c RET
sort lines in region                              C-c ^

-------------------------------------------------------------------------------
Regions
-------------------------------------------------------------------------------

cut/copy/paste rectangular region                 C-c C-x C-w/M-w/C-y
fill paragraph across selected cells              C-c C-q

-------------------------------------------------------------------------------
Miscellaneous
-------------------------------------------------------------------------------

to limit column width to N characters, use        ...| <N> |...
edit the current field in a separate window       C-c `
make current field fully visible                  C-u TAB
export as tab-separated file                      M-x org-table-export
import tab-separated file                         M-x org-table-import
sum numbers in current column/rectangle           C-c +

-------------------------------------------------------------------------------
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 ~
```

```
------------------------------------------------------------------------------
Spreadsheet
------------------------------------------------------------------------------

Formulas typed in field are executed by TAB,
RET and C-c C-c.  = introduces a column
formula, := a 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 2nd to 3rd hline                           |:=vsum(@II..@III)|
apply current column formula                        | = |

set and eval column formula                         C-c =
set and eval field formula                          C-u C-c =
re-apply all stored equations to current line       C-c *
re-apply all stored equations to entire table       C-u C-c *
iterate table to stability                          C-u C-u C-c *
rotate calculation mark through # * ! ^ _ $         C-#
show line, column, formula reference                C-c ?
toggle grid / debugger                              C-c }/{

------------------------------------------------------------------------------
Formula Editor
------------------------------------------------------------------------------

edit formulas in separate buffer                    C-c '
exit and install new formulas                       C-c C-c
exit, install, and apply new formulas               C-u C-c C-c
abort                                               C-c C-q
toggle reference style                              C-c C-r
pretty-print Lisp formula                           TAB
complete Lisp symbol                                M-TAB
shift reference point                               S-cursor
shift test line for column references               M-up/down
scroll the window showing the table                 M-S-up/down
toggle table coordinate grid                        C-c }

==============================================================================
Links
==============================================================================

globally store link to the current location         C-c l \noteone
insert a link (TAB completes stored links)          C-c C-l
insert file link with file name completion          C-u C-c C-l
edit (also hidden part of) link at point            C-c C-l
```

```
open file links in emacs                              C-c C-o
...force open in emacs/other window                   C-u C-c C-o
open link at point                                    mouse-1/2
...force open in emacs/other window                   mouse-3
record a position in mark ring                        C-c %
jump back to last followed link(s)                    C-c &
find next link                                        C-c C-x C-n
find previous link                                    C-c C-x C-p
edit code snippet of file at point                    C-c '
toggle inline display of linked images                C-c C-x C-v

================================================================================
Working with Code (Babel)
================================================================================

execute code block at point                           C-c C-c
open results of code block at point                   C-c C-o
check code block at point for errors                  C-c C-v c
insert a header argument with completion              C-c C-v j
view expanded body of code block at point             C-c C-v v
view information about code block at point            C-c C-v I
go to named code block                                C-c C-v g
go to named result                                    C-c C-v r
go to the head of the current code block              C-c C-v u
go to the next code block                             C-c C-v n
go to the previous code block                         C-c C-v p
demarcate a code block                                C-c C-v d
execute the next key sequence in the code edit bu     C-c C-v x
execute all code blocks in current buffer             C-c C-v b
execute all code blocks in current subtree            C-c C-v s
tangle code blocks in current file                    C-c C-v t
tangle code blocks in supplied file                   C-c C-v f
ingest all code blocks in supplied file into the      C-c C-v i
switch to the session of the current code block       C-c C-v z
load the current code block into a session            C-c C-v l
view sha1 hash of the current code block              C-c C-v a

================================================================================
Completion
================================================================================

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

complete word at point                                M-TAB
```

```
================================================================================
TODO Items and Checkboxes
================================================================================

rotate the state of the current item              C-c C-t
select next/previous state                        S-LEFT/RIGHT
select next/previous set                          C-S-LEFT/RIGHT
toggle ORDERED property                           C-c C-x o
view TODO items in a sparse tree                  C-c C-v
view 3rd TODO keyword's sparse tree               C-3 C-c C-v

set the priority of the current item              C-c , [ABC]
remove priority cookie from current item          C-c , SPC
raise/lower priority of current item              S-UP/DOWN\notetwo

insert new checkbox item in plain list            M-S-RET
toggle checkbox(es) in region/entry/at point      C-c C-x C-b
toggle checkbox at point                          C-c C-c
update checkbox statistics (C-u : whole file)     C-c #

================================================================================
Tags
================================================================================

set tags for current heading                      C-c C-q
realign tags in all headings                      C-u C-c C-q
create sparse tree with matching tags             C-c \\
globally (agenda) match tags at cursor            C-c C-o

================================================================================
Properties and Column View
================================================================================

set property/effort                               C-c C-x p/e
special commands in property lines                C-c C-c
next/previous allowed value                       S-left/right
turn on column view                               C-c C-x C-c
capture columns view in dynamic block             C-c C-x i

quit column view                                  q
show full value                                   v
edit value                                        e
next/previous allowed value                       n/p or S-left/right
edit allowed values list                          a
make column wider/narrower                        > / <
```

```
move column left/right                              M-left/right
add new column                                      M-S-right
Delete current column                               M-S-left


================================================================================
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 / d
the time between 2 dates in a time range            C-c C-y
change timestamp at cursor A+/-1 day                  S-RIGHT/LEFT\notetwo
change year/month/day at cursor by A+/-1              S-UP/DOWN\notetwo
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
select date while prompted                          mouse-1/RET
toggle custom format display for dates/times        C-c C-x C-t


--------------------------------------------------------------------------------
Clocking time
--------------------------------------------------------------------------------

start clock on current item                         C-c C-x C-i
stop/cancel clock on current item                   C-c C-x C-o/x
display total subtree times                         C-c C-x C-d
remove displayed times                              C-c C-c
insert/update table with clock report               C-c C-x C-r

================================================================================
Agenda Views
================================================================================

add/move current file to front of agenda            C-c [
remove current file from your agenda                C-c ]
cycle through agenda file list                      C-'
set/remove restriction lock                         C-c C-x </>

compile agenda for the current week                 C-c a a \noteone
compile global TODO list                            C-c a t \noteone
compile TODO list for specific keyword              C-c a T \noteone
match tags, TODO kwds, properties                   C-c a m \noteone
```

```
match only in TODO entries                          C-c a M \noteone
find stuck projects                                  C-c a # \noteone
show timeline of current org file                    C-c a L \noteone
configure custom commands                            C-c a C \noteone
agenda for date at cursor                            C-c C-o

-------------------------------------------------------------------------------
Commands available in an agenda buffer
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
View Org file
-------------------------------------------------------------------------------

show original location of item                       SPC/mouse-3
show and recenter window                             L
goto original location in other window               TAB/mouse-2
goto original location, delete other windows         RET
show subtree in indirect buffer, ded.\ frame         C-c C-x b
toggle follow-mode                                   F

-------------------------------------------------------------------------------
Change display
-------------------------------------------------------------------------------

delete other windows                                 o
view mode dispatcher                                 v
switch to day/week/month/year/def view               d w vm vy vSP
toggle diary entries / time grid / habits            D / G / K
toggle entry text / clock report                     E / R
toggle display of logbook entries                    l / v l/L/c
toggle inclusion of archived trees/files             v a/A
refresh agenda buffer with any changes               r / g
filter with respect to a tag                         /
save all org-mode buffers                            s
display next/previous day,week,...                   f / b
goto today / some date (prompt)                      . / j

-------------------------------------------------------------------------------
Remote editing
-------------------------------------------------------------------------------

digit argument                                       0-9
change state of current TODO item                    t
kill item and source                                 C-k
archive default                                      $ / a
refile the subtree                                   C-c C-w
set/show tags of current headline                    : / T
```

```
set effort property (prefix=nth)                  e
set / compute priority of current item            , / P
raise/lower priority of current item              S-UP/DOWN\notetwo
run an attachment command                         C-c C-a
schedule/set deadline for this item               C-c C-s/d
change timestamp one day earlier/later            S-LEFT/RIGHT\notetwo
change timestamp to today                         >
insert new entry into diary                       i
start/stop/cancel the clock on current item       I / O / X
jump to running clock entry                       J
mark / unmark / execute bulk action               m / u / B

--------------------------------------------------------------------------------
Misc
--------------------------------------------------------------------------------

follow one or offer all links in current entry    C-c C-o

--------------------------------------------------------------------------------
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

================================================================================
LaTeX and cdlatex-mode
================================================================================

preview LaTeX fragment                            C-c C-x C-l
expand abbreviation (cdlatex-mode)                TAB
insert/modify math symbol (cdlatex-mode)          ` / '
insert citation using RefTeX                      C-c C-x [

================================================================================
Exporting and Publishing
================================================================================
```

```
Exporting creates files with extensions .txt and .html
in the current directory.  Publishing puts the resulting file into
some other place.

export/publish dispatcher                          C-c C-e

export visible part only                           C-c C-e v
insert template of export options                  C-c C-e t
toggle fixed width for entry or region             C-c :
toggle pretty display of scripts, entities         C-c C-x {\tt\char`\}

------------------------------------------------------------------------------
Comments: Text not being exported
------------------------------------------------------------------------------

Lines starting with # and subtrees starting with COMMENT are
never exported.

toggle COMMENT keyword on entry                    C-c ;

==============================================================================
Dynamic Blocks
==============================================================================

update dynamic block at point                      C-c C-x C-u
update all dynamic blocks                          C-u C-c C-x C-u

==============================================================================
Notes
==============================================================================
[1] This is only a suggestion for a binding of this command.  Choose
your own key as shown under ACTIVATION.

[2] Keybinding affected by org-support-shift-select and also
 org-replace-disputed-keys.
```

Tags

There are two types of tags, `todo` tags and `:patt1:` tags, working with them is essentailly the difference between using `t` and `m` respectively.

There is a tag line at the top of the file that lists tags included in this file, and you can add tags to the `org-tag-alist` variable, this is worth doing to prevent you from creating double tags.

You can also merely specify any tag in the headline using `:patt1:` which can be triggered by

using C-c C-q, it's probably better to specify tags in the preamble and/or variable because you can set mutually exclusive and nested tags. after doing this make cure to press C-c C-c on the #+TAGS: line in order to activate the changes

DONE Listing used tags tagging

<2019-11-19 Tue>

you should listt all tags like this to add them to the persistent list:

this is a drawer

I use the following chunk of elisp to create a dynamic block containing a table where each row is a tag and number of use of that tag in the file. Add the block to your org file and hit C-c C-c to expand it:

```
#+BEGIN: tagblock
#+END:
```

If you only want to count certain tags you can add a :tags option to the block and to add a label to the resulting table use :label

```
#+BEGIN: tagblock :tags ("tag1" "tag2" "tag3") :label tag-counts
| tag1 | 2 |
| tag2 | 4 |
| tag3 | 8 |
#+END:
```

With the :todo option, only entries with a todo state (either any or in a list you specify) will be counted.

All the real work is done by the function org-freq-count which takes a parameterized search string (the same as an agenda tags search) and replaces %s in the search with each element of targets, counting the number of matches. If cmp is given, it will be used as a comparison function when sorting the output.

```
(defun org-freq-count (search targets &optional cmp)
  (let ((cmp (if (functionp cmp)
                 cmp
               (lambda (a b) nil))))

    (mapcar (lambda (x)
              (list x (length (org-map-entries t (format search x) nil))))
            (sort
             (delete-dups
              (-filter #'stringp targets))
             cmp)
            )
    ))

(defun org--tagblock-all-tags ()
  (-filter #'stringp (-map #'car (append
                                  (org-get-buffer-tags)
```

```
                                    org-tag-alist
                                    org-tag-persistent-alist))))

(defun org-write-freq-count (search targets name)
  (insert (s-concat
           (if name (insert (format "#+NAME: %s\n" name)))
           (mapconcat
            (lambda (x) (format "| %s | %s |" (nth 0 x) (nth 1 x)))
            (org-freq-count search targets)
            "\n")))
  (org-table-align)
  )

(defun org-dblock-write:tagblock (params)
  (let ((todo (plist-get params :todo))
        (tags (or (plist-get params :tags) (org--tagblock-all-tags)))
        (label (plist-get params :label))
        (caption (plist-get params :caption))
        )
    (when caption (insert (format "#+CAPTION: %s\n" caption)))
    (org-write-freq-count (cond ((equal todo t)
                                 (format "%%s/%s" (mapconcat 'identity
                                                             org-not-done-keywords
                                                             "|"
                                                             )))
                                ((listp todo)
                                 (format "%%s/%s" (mapconcat 'identity
                                                             todo
                                                        "|"
                                                        ))
```

Agenda Searching agenda:tagging

Basically the search attaches only to things beneath a headline, like a block. To search for multiple terms you have to put + between the tearms, or you can put this in your `init.el`

```
(setq org-agenda-search-view-always-boolean t)
```

Then search the agenda with org-agenda which is:

- SPC o a in Doom
- SPC a o in Spacmacs

After getting a result of tags you can further filter the tags by using backslash \ (and maybe TAB on Doom) to further match tags

Frustratingly this isn't recursive like my bash script here but the layout is such that you can see what's going on if your eyes are good.

Inline Code

First Steps

First you have to activate languages, this is what I used:

```
;; Active Babel languages
;; Active Babel languages
(org-babel-do-load-languages
   'org-babel-load-languages
    '((R          . t)
      (latex      . t)
      (python     . t)
      (gnuplot    . t)
      (java       . t)
      (sed        . t)
      (shell      . t)
      (mathematica . t)
      (emacs-lisp  . t)))
```

You need to specify how you want the code block to behave:

- `:exports code`
  - use this if you don't want the code evaluated (but want highlighting)
- `:  exports both`
  - Code listing and result
- `:exports results`
  - Only the results of the code
- `:exports none`
  - The code just doesn't get exported

If you don't want something to evauluate set `:exports none`

Execute External Language

External languages can be executed by pressing `C-c C-c` and then pressing `y` for yes

```
print(rnorm(3))
```

-0.189924291082191
0.399005404851512
-1.71807612110756

This can be handy for complicated things like:

```
words <- tolower(scan("1.org", what="", na.strings=c("|",":")))
t(sort(table(words[nchar(words) > 3]), decreasing=TRUE)[1:3])
```

that	this	with
129	110	105

Working with R R:diagrams

Integrating ESS Auto Complete R:AutoComplete

first you need to add ESS to the spacemacs layers[5].

1. Using ESS once it's installed load any `.R` file (preferably in a new frame with `C-x 5 C-f` and it should just work[tm] and be in `ESS[R]` mode, if not activate that mode by using `M-x r-mode`.

   Next you will want to opon a REPL console underneath, which should be in `iESS` mode with the name **R**, do this by using `M-x R`, it will ask for a project location / directory, this is annoying and 90% of the time you're happy to have that as the working directory location of the original file, in order for that to happen put the following in your `.emacs.d/init.el`:

   ```
   ;R Binding for <-
   (setq-default dotspacemacs-configuration-layers '((ess :variables
                                                          ess-assign-k␣
                                                          ↪ ey
                                                          ↪ "\M--")))

   ;Set starting project directory to location of buffer =M-x pwd=
   ;https://ess.r-project.org/Manual/ess.html#Customizing-startup
   (setq ess-ask-for-ess-directory nil)
   ```

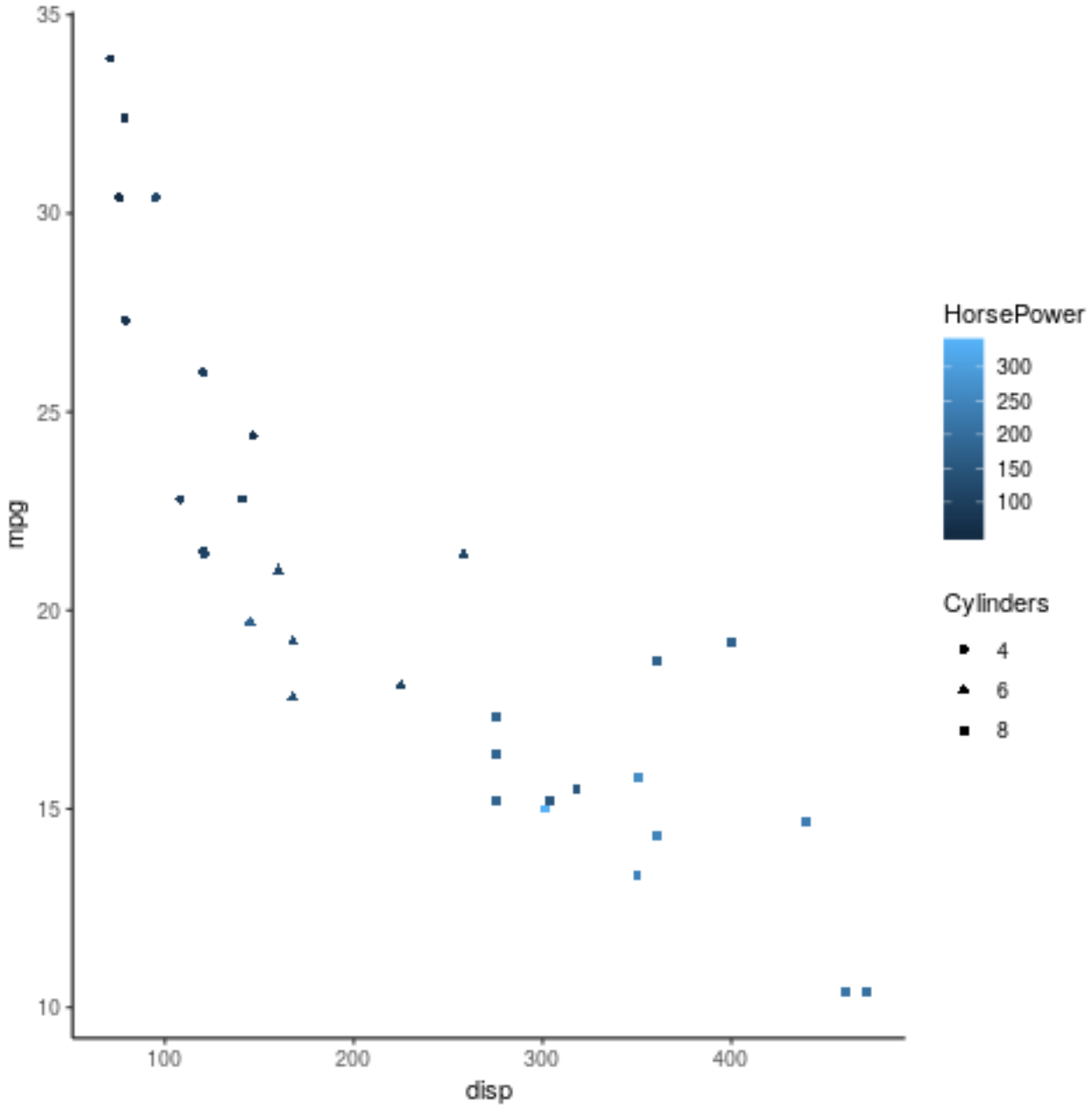
   if you want to open a seperate frame use `SPC w F` (remember a frame is an external X window and a window inisde by the Emacs language) and if you want to have the source code scroll over multiple panes use `follow-mode` by typing `SPC w f`

[5] Look at the *GitHub* for the ESS layer

2. Using ESS inside Org So one way is to just open an indirect buffer and/or change the mode, but that's a little painful because sometimes the point will be moved away and `C-o / g;` don't always work.

   another way is to switch to the buffer for that chunk using `org-edit-special` which is mapped to `C-c ' / , '` , this works for other environments (read `C-h f org-edit-special`), when it's used in src code it will take copy code to a seperate buffer and when you finish by using `M-Ret '` it will copy it back, so here I've used it to make a plot of *MtCars*:

```
library(tidyverse)
mtcars <- as_tibble(mtcars)
myplot <-  ggplot(mtcars, aes(x = disp, y = mpg, col = hp, shape =
↪  as.factor(cyl))) +
  geom_point() +
  theme_classic() +
  labs(col = "HorsePower", shape = "Cylinders")
myplot
```



Make sure to use the header arguments as shown above, there was a change to the `org-mode` version that broke this recently, now `results: ouput graphics file` is specifically required.[6]

(a) Naming plots ATTACH Unfourtunately, every single code block MUST be named in order for this to work, what can happen is that using pandoc to export `*.rmd` –> `*.md` includes a link to a path of an image and you mistake that as a `#+RESULTS:`, it will even appear under the `#+RESULTS:` because the results are blank, blank because the function gives no output, but, that's still just a link to a file and nothing is generated, easy mistake to make.

[6] https://stackoverflow.com/a/60070347/12843551

You can set a global setting using

```
```

but if you set `:file temp.png` globally every file will overwrite every other file, this is fairly undesirable, so instead use something like this:

```
(defconst our-charset "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabc defghijklmnopqrstuvwxyz")
(defconst our-charset-length (length our-charset))

(defun random-string (&optional max-length min-length)
  "Generate a random string."
  (let (string)
    (dotimes (_i (+ (random (- (or max-length 10) (or min-length 5) -1)) (or min-length 5)))
      (push (aref our-charset (random our-charset-length)) string))
    (concat string)))

(random-string)
```

And then just feed that to the function:

And so then the idea is that you would just specify that in the header with something like this:

```
```

This however has the downside that EVERYTHING evaluated will be exported unfourtunately.

Now the problem with this is that if you have similar file names in the same directory it won't work, an easy solution is either generating arandom-string or using the file name The best about this though is that you can even split the window up to get an ***R*** `iESS` console underneath, follow mode on the right and the `org-edit-src-code` in one of the corners.

(b) Working Directories Strangely `setwd()` won't persist in `org-mode`, instead it is necessary to set the working directory for every code block, this clearly isn't ideal so instead you can use:

```
#+PROPERTY: header-args:R  :session *R* :dir "../"
```

actually don't use this either, this causes problems as well, tread `org` just like `rmd` and always have the file in it's working directory, otherwise images that get put together upon export don't work properly.

Plotting

You can produce plots as well, but what gets complicated is using the correct options in the header, they are reasonably self explanatory though, for example the following code: (You need to include `output` if

you want to use ggplot2 etc., look at this answer and the docs)

```
#+BEGIN_SRC R :cache yes :exports both :results output graphics :file test.png
  library("ggplot2")
  ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length, color = Species)) +
  geom_point() +
  theme_bw()
\#+END_SRC
```

will produce the following plot and list the code appropriately:

```
1  library("ggplot2")
2  ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length, color = Species)) +
3  geom_point() +
4  theme_bw()
```

DONE Is there an easy way to convert rmd to org?

<2019-11-13 Wed>

DONE Syntax Highlighting

By default the r-code doesn't have synatx highlighting. In order to get syntax highlighting in LaTeX export you need to either use the *minted* or *listings* package, to use the minted package specify the following in your `~/.emacs.d/init.el`:

```
1  (require 'org-latex)
2  (setq org-export-latex-listings 'minted)
3  (add-to-list 'org-export-latex-packages-alist '("" "minted"))
```

Would it be worth using org-mode to manage code notes over boostnote?

Because boostnote is just open source `*.md` files, switching out is no drama, given how good `org-mode` is i'd probably be better off just abandoning *Notable* all togethr and using *Boostnote* for want of the features (and hey open source) and use org-mode for everything else.

1. Using Org-Mode to replace Boostnote

Pros	cons
+ Closer to Terminal and Code	+ Less User Friendly
+ Easier to switch into because always open and bookmarks/recents	+ tags aren't as nice/usable
+ Way better text editing	
+ Can open fles externally	
+ Brings all notes together	

DONE How do I include ggplot inline? R

1. ESS or R-org A really good tutorial on using ***R*** in org-mode is org-mode *Worg* tutorials

DONE Java Example

Write an example of java code

```
// This is my first Java Program.
  public class MyFirstProg{
      public static void main(String[] args) {
          System.out.println("I'm a Program!");
      }
  }
```

Syntax Highlighting

Syntax highlighting corresponds to major modes in vim, for example there is a `python-mode`, because the word `python` is suffixed with suffixed with `-mode`[7] a `python` block shoudl be started by using `#+begin_src python`, like wise we would use `R-mode` to edit ***R***, `vim` is a bit of an exception in this way, because highlightJS refers to it as `vimscript` whereas *Emacs* uses `vimrc`[8].

It's important to only evel call it `vim`, because the `minted` package in LaTeX will fail if it does not recognize the language, instead the appropriate avenue is instead to have `org` accept that we need to call the environment `vim` even if the mode is `vimrc-mode` [9]:

```
(add-to-list 'org-src-lang-modes (cons "vim" 'vimrc))
```

If you do use `#+begin_src vimrc` there should be syntax highlighting inside `org-mode` and inside the export.

DONE Including Tikz Plots diagrams:Buffers

CLOSED: *[2019-11-13 Wed 11:53]*

[7] [[[emacs - Syntax highlighting in org-mode. List of languages? - Stack Overflow](https://stackoverflow.com/a/22484646)][]]

[8] GitHub - mcandre/vimrc-mode: Enables syntax highlighting for .vimrc/vimrc files

[9] emacs - Syntax highlighting in org-mode. List of languages? - Stack Overflow

DONE Using a tikz picture

I could only get this to work in HTML following the example from UCL there's reference to this orgmode.org/worg tutorial on LaTeX in org where the idea is:

- If it's exported to HTML
  - then convert the tikz to zvg and insert the svg's
- else if it's exported to LaTeX
  - just include the LaTeX
- else just leave it raw
- the `t` backend refers to the inline org-mode *toggling*
  - I just remember it as *toggling* because I can find no documentation on it.
  - it is always triggered so you must put it at the end or it won't work properly

but it only works in HTML so far. The `+#_Header: = arguments are the same as putting them on the line =#+BEGIN_SRC`, but that would be a really long line.

In order to use the if then statements as I have add the following to the `~/.emacs.d/init.el` file:

```
(setq org-babel-latex-htlatex "htlatex")
(defmacro by-backend (&rest body)
  `(case (if (boundp 'backend) (org-export-backend-name backend) nil)
    ,@body))
```

Making it work across formats

So the trick is, execute this tikz like this with `C-c C-c`:

```
#+LATEX_HEADER: \usepackage{tikz}
#+HEADER: :headers '("\\usepackage{tikz}")
#+HEADER: :fit yes :imoutoptions -geometry 400 :iminoptions -density 600
#+header: :file (by-backend (html "./img/dosage.svg") ) :imagemagick yes
#+header: :results (by-backend (pdf "latex")  (html "raw") (t "raw"))
\#+BEGIN_SRC latex  :file (by-backend (t "./img/dosage.svg") ) :imagemagick yes
\begin{tikzpicture}[domain = 0:10, scale = (2/3)]
  \clip (-1,-1) rectangle (12,12);
  \draw[->, thick] (0,0) -- (0,10) node[right] {$C$ {\scriptsize nmol $\cdot  $ L$^{-1}$}};
  \draw[->, thick] (0,0) -- (10, 0) node[right] {$t$ };
  \draw[] [out=270, in = 180]  (0,5) to (3,2);
  \draw[dashed] (3,2)--(3,5);
  \draw[] [out=270, in=180] (3,5) to (6,3) ;
  \draw[dashed] (6,3)--(6,6);
  \draw[] [out=270, in=180] (6,6) to (9,4) ;
```

```
  \draw[dashed] (9,4)--(9, 7);
  \draw[ ] [out=270, in=180] (9,7) to (12,4);
  \draw[dotted]  (0,7)--(12,7) node[below left] {$C_n = H$};
  \draw[dotted]  (0,4)--(12,4) node[below left] {$R_\infty = L$};
  \node [left] at (0,5) {$C_0$};
  \draw[<-> ] (0.4,4)--(0.4,7) node[below right] {\tiny $C_0 = H-L$};
\end{tikzpicture}
\#+END_SRC
```

Then that should create, below it a section that looks like this:

```
#+RESULTS:
[[file:./img/dosage.svg]]
```

This results section will inline preview but it won't get pulled in by an export process, so now that the inline preview is done we can take the line immediately preceeding the code fence and ~~'comment it out'~~ delete it (comments, either `#_` or = or whatever DO NOT WORK between header and source so do not use it. you should delete the line after `latex` that matches:

```
:file (by-backend (t "./img/dosage.svg") ) :imagemagick yes
```

And then we have tikz that compiles in LaTeX and an SVG that will 'just work^{tm}' when exporting to HTML and that can be leveraged for other exports.

Example Tikz Export **Tikz**

```
\begin{tikzpicture}[domain = 0:10, scale = (2/3), draw = purple, text= pink]
  \clip (-1,-1) rectangle (12,12);
  \draw[->, thick] (0,0) -- (0,10) node[right] {$C$ {\scriptsize nmol $\cdot  $ L$^{-1}$}};
  \draw[->, thick] (0,0) -- (10, 0) node[right] {$t$ };
  \draw[] [out=270, in = 180]  (0,5) to (3,2); % node[right] {\scriptsize \(\left( t_{\textit{r
  \draw[dashed] (3,2)--(3,5);
  \draw[] [out=270, in=180] (3,5) to (6,3) ;
  \draw[dashed] (6,3)--(6,6);
  \draw[] [out=270, in=180] (6,6) to (9,4) ;
  \draw[dashed] (9,4)--(9, 7);
  \draw[ ] [out=270, in=180] (9,7) to (12,4);
  \draw[dotted]  (0,7)--(12,7) node[below left] {$C_n = H$};
  \draw[dotted]  (0,4)--(12,4) node[below left] {$R_\infty = L$};
  \node [left] at (0,5) {$C_0$};
  \draw[<-> ] (0.4,4)--(0.4,7) node[below right] {\tiny $C_0 = H-L$};
\end{tikzpicture}
```

and it will create below something that says `#+RESULTS: file:./img` which should also inline display the image, now you can freely delete everything after `latex` in the header and it will work in all exports and you'll get to have the inline display, just copy it back if you want it again.

Example 2

1. Snippet SNIPPETS Using the tutorial here and the *YaSnippets* manual I turned this into a snippet in orgmode/plot. The snippet can be triggered by typing in `plot` and positinioning the cursor over and then running `M-x yas-expand` / `:yas-expand` / `<SPC> <SPC> yas-expand`.

   - If the minor mode is enabled with `M-x yas-minor-mode` then I think you could just press `TAB` / `M-/` after plot and then jumping to the next field with `M-s-/`
   -

2. Exporting Guidelines If the svg files get deleted, they won't show up in exports, but they can be regenerated by moving the line back up and recompiling the tikz, the thing is, this code was intended to be fool-proof automatic, but it's not, and it might be more complicated that it needs to be, maybe just let sleeping dogs lie though.

TikzJaz

You could have the HTML deal with the raw TikZ, this would be way simpler and something I should consider, all I would need to do is add a few lines to the header of a HTML, which I could achieve with `#+HTML_HEADER:`, a corresponding package is TikzJax, I'm not going to pursue this for the moment though because if I need to, on occasion, export to markdown or ODT, I'll be stuck without SVGs.

Set up a workflow

Images

- How do I do screenshots
- How do I do ipad drawings?
- How do I do drawings from inkscape
- How do I just do outright tikz?

DONE Install the /Prelude/Starter Kit.

DONE How do I deal with headlines?

use `M-left/right` to demote and promote them, use `M-up/down` to move them. To do items may be cycled by using =S-left/right.

Moving Headlines

Demoting and promoting headlines

DONE Can I Hide all the ToDo Tags from export?

Let's say that I use the todo tags to manage my notes and dates etc. Could I hide that from an export? It looks like no but I could filter the tree and just toggle the state with `t`, export it, and not save the buffer, a hacky work arround but sufficient for now, I could also just use `regex`

DONE Using Emacs for Notes

DONE Browsing all ToDO item Todo

If you've added a file to the agenda-variable with `C-c [` all todo items in the agenda can be searched using `<Spc> m a t`

DONE Navigating a file

How do I jump to the top of a headline so that I can use `TAB` to fold it ? You can use `C-c C-f=/=C-c C-n` / `gj` / `gl` will go forward headlines and `C-c C-b` / `gk` / `g;` / `C-c C-p` will go back headlines. I have remapped these to `z k` and `z j` respectively

1. Folding How do I fold and unfold the entire document?
2. Marking Location Can I still use `m CHAR` to mark a location and jump back to it?
3. Frames and windows often you will want to use `<SPC> m b | , b` to open an indirect buffer for items beneath a headline, it will then open a second window below that is out of focus `C-w C-w` / `C-x o` / `<SPC> w w` / `M-m w w` can be used to pull that window into focus (or any window with `SPC <NUM>` and then that window can be maximized with `C-x 1` / `<SPC> w m`
4. Finding Shortcuts BUFFERS If you're using spacemacs and you know a single shortcut it isn't usually difficult to find more shortcuts:
   (a) Switch to fullscreen
   (b) Move to a headline and press `, b`
   (c) Slowly enter the key-sequence `C-x`
   (d) Observe that the option `o` corresponds to the function `maximize-buffer`
   (e) Press `C-g` to clear out of the *minibuffer*
   (f) Press `<SPC> <SPC>` / `M-x` and type in `maximize-bu`
   (g) At or around the top should be the text `spacemacs/toggle maximize-buffer (M-m w m)`
   (h) Usually spacemacs will map `SPC` to `M-m` and in this case the corresponding shortcut is indeed `<SPC> w m`, if however it wasn't there's *Google* because you have the corresponding command.

It's also possible to switch buffers with `<SPC> b n`.

Listing buffers is weird, if you use `C-x b` / `:ls` you can just start typing in the file name, but if you get up the list of buffers with `C-x C-b` it won't take focus which is really annoying.

press `<SPC> <SPC>` / `M-x`

DONE Attaching files **Attachments**

Attachments are really cool, they can be copied or symlinked using `<SPC> m i a`, that's even cause to give up on dropbox, the file can also just be linked to.

If you want to just link the file, open the org in vim and then use `fzf`.
fzf can be enabled in spacemacs using the AshyIsMe/fzf-spacemacs-layer

DONE Reading an Emacs File

- How do I fold a section rather than the whole thing?
- To fold the whole thing I can use `S+TAB+` which is analogous to `zr` and `zm` in vim but it cycles through

DONE Use Vim Can I create a keybinding to open the file or buffer in *Vim*? **vim**

DONE How to mark lines just like in vim?

Using the bookmarks shortcuts are amazing, refer to them in*Spacemacs* Documentation, basically hit `<SPC> f b` and you can create and jump to bookmarks across and within files.

- Bookmarks can be deleted with ,[10]
  - `C-d`
- Opened in another frame (i.e. an entirely new X Window)
  - `C-c C-o`
- Opened in anotgher window (i.e. internal to *Emacs*
  - `C-c o`

DONE Opening Current File in vim

- Is there a way to bring my bindings into space macs?
  - Probably, but it's a lot of work when you can just use both programs with no loss.

[10] Helm-bookmark manual

The binding (`C-c o` in *Prelude* and `Spc f 0` in *Spacemacs* will open the current file in an external editor (I'm not sure if the file or the buffer tbh) just make sure the system has default editor as gvim and that *gvim* is set to read the correct vimrc and your set, just remember to call `PlugUpdate`. Pure vim stopped working for me recently, I think it's getting trapped by the emacs terminal

I've found two solutions for opening the current buffer in vim, on on the *Emacs Wiki* which causes *Emacs* to crash and another one on *Stack Exchange* that works flawlessly and remembers the line number:

```
(defun my-open-current-file-in-vim ()
  (interactive)
  (async-shell-command
   (format "gvim +%d %s"
       (+ (if (bolp) 1 0) (count-lines 1 (point)))
       (shell-quote-argument buffer-file-name))))
```

If you want to use kitty terminal just specify `kitty` and avoid the popup [11] change it to `call-process-shell-comma`

```
(defun my-open-current-file-in-vim ()
 (interactive)
 (call-process-shell-command
                                      ;  (format "gvim +%d %s"
  (format "~/.local/kitty.app/bin/kitty -e nvim +%d %s"
          (+ (if (bolp) 1 0) (count-lines 1 (point)))
          (shell-quote-argument buffer-file-name))))
```

Maybe later having something for notepad/vim etc. would be nice but it's pretty simple to just open stuff from vim win `:gedit "%"` and remapping that with `autocmd BufEnter *.org :map <f12> :w<cr>:!gedit "%" <Enter>`.

Actually it might have been `C-u C-c o`

Also to open a new line (i.e. `o` in vim) use `S-Enter`

DONE Navigating Files

1. **DONE** Moving to a previous location If you are following an internal link (i.e. a link to a target in the org-mode file), then org-mode automatically pushes the position in the mark-ring before jumping.

   So, you follow a link with `C-c C-o`, then you can jump back with `C-c &`.[12]

2. Recently closed Use the key sequence `SPC f r` for **F*ind *R*ecent* file.

[11] asynchronous - How to avoid pop-up of **Async Shell Command** buffer in Emacs? - Stack Overflow

[12] Refer to the documentation here.

DONE Exporting Style

DONE In Line References

In order to reference code blocks using org-ref you need to install the `org-ref` package by following the instructions on their GitHub, you need to add a line to your `.emacs.d/.init.el` but then you should be able to do `M-x package-list-packages` and find `org-ref` in order to install it.

According to this post you need to set `org-latex-prefer-user-labels` to `non-nil` in order to use the custom ID given in `#+NAME`. Refer to the help with `C-h v org-latex-prefer-user-labels` and the setting can be toggled from there, the reason for the default is if you multiply define a label or use wrong syntax the latex won't generate.

I can't find a simple way to easily reference figures and source code cross-reference wise or academic wise, this could take a very long time.

1. Dedicated targets maybe an easier way to figure this out is by using a Dedicated target (go to top), and you can reference sections by using `[[Section Name]]`
2. **DONE** I haven't figured out how to fix labels with HTML though I tried looking for `org-HTML-prefer-user-labe` but no luck, there might be something in the Manual.

DONE Markdown

I need to figure how to export as markdown without breaking the dam thing

DONE Export as eMail with Math

The easiest way is to take the exported html and download it using an add on (*SingleFile* Export) and then copy that html, which is now mathml, into thunderbird, trying to convert mathjax to mathml without losing the inline css is like trying to pull out teeth.

Telling Org to use math-ml doesn't work, Using SVG's don't get embedded inline, and, I don't really want to use pandoc when I don't have to given that the inline is so nice, if I did however I could probably do something like this:

```
pandoc -s --self-contained input.org --toc  -c
  ↪  ~/.emacs.d/org-css/Killercup.css -o out.html
```

1. Test Email So the quadratic formula is given by:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{5}$$

   the geometric series is given by

$$S_n = \sum_{i=0}^{n-1} [a \cdot r^n] \tag{6}$$

$$= \frac{1 - r^n}{1 - r} \tag{7}$$

$$\implies \sum_{n=0}^{\infty} = \frac{1 - \lim_{n \to \infty} [r^n]}{1 - r} \tag{8}$$

DONE Styles

1. **DONE** HTML Styles I found this nice Trick on Reddit (insert a link with `C-c C-l`)[13], in order to have inline CSS upon export (rather than two files) put CSS files in `~/.emacs.d/org-css/` and `run =toggle-org-custom-inline-style` in an org buffer associated with a file, it should now prompt you for a theme on the next HTML export if you use this code shown in **??** [14]

   So I did an error once where it complained about citeproc being an invalid function, ifthat occurs just use `M-x package-install-file` to install `~/Dropbox/DotFiles/Spacemacs/Downloads/citeproc-org-` and that seems to just about fix it.

2. done LaTeX Styles upon Export. I can, I believe, set up a custom package for a template (of course that means that I'll have to fix the style that I've been using from Computer algebra) by specifying a header of the form `#+LATEX_HEADER: \usepackage{mystyle.sty}`.

   I need to fix my LaTeX file so it will play ball externally

   When exporting to latex you need to be careful with loading in svg's, you will need to load a corresponding package

   In order to enable the listings package in a pdf export use the following in your `~/.emacs.d/init.el`:

   ```
   (require 'ox-latex)

   (setq org-latex-listings t)
   (add-to-list 'org-latex-packages-alist '("" "listings"))
   (add-to-list 'org-latex-packages-alist '("" "color"))
   ```

   It's also important never to include footnotes or hyperlinks in headings, even math should be avoided because LaTeX will throw non-descriptive errors, also I believe you need to be careful using the word `LaTeX` and this should perhaps be enclosed in code tags or deilberately lowercased.

   the command `\LaTeX` should never be wrapped in `$$` or `\( \)` because otherwise LaTeX will crash and because it automatically changes upon export..

   Never include multi-line math in tables, that's another thing latex really doesn't like

[13] The shortcut for footnotes is `C-c C-x f` as described in the manual and the shortcut for links is `C-c C-l`

[14] having `emacs-lisp` after means it will be (atleast-partially) evaluated upon HTML export, you will need to install `emacs-htmlize` in order to export it, this can be done by typing in the following `M-x package-install RET htmlize`; confusing that it isn't `emacs-htmlize`, maybe that's a standard?

Try to avoid unicode because while it can maybe work with XeLaTeX, it doesn't work with listings and is simply not worth the headache.

3. JavaScript A tonne of options for exporting with extended javascript are also at my disposal, look at the Manual, most importantly is `view:content` and `view:info`, so for example:

   ```
   view:info toc:nil
   #+INFOJS_OPT: view:info toc:nil
   ```

   Once your in the html, use `i` to toggle the *Table of Contents*.

   - How to specify a template for LaTeX

   So turn my typical `LaTeX` template into a `\usepackage{}`

4. **DONE** Create CSS files Take the CSS files and merge them with something bare bones to colour the ToDO tags

5. Shortcuts Exporting can be performed by using the shortcut `<SPC> m e e=/=C-c C-e`.

6. **DONE** use spacemacs WORKFLOW when using spacemacs the config file remains as `~/.emacs.d/init.el`, the default config for all *spacemacs* settings however becomes the `~/.spacemacs` file, which, by default, has all the default settings in it; it is important to note that spacemacs settings will only be respected in that template and in that very specific style of wrapping, so just open the file and edit the settings of the `elisp` in place. it's well worth Reading the Documentation because it also mentions things like support for `GitHub Flavoured Markdown`

7. Changing Themes You can install a bunch of themes by installing the `themes-megapack` as documented on the GitHub repo, but the only way to change a theme that I've found is to use `<M-x spacemacs/helm-themes>`.

   You'll need to change the default theme at loading however, because, the $LaTeX$ will come out wrong otherwise, like black on a white background etc.

   If however you go to the `.spacemacs` file and locate the string `List of themes` there will be a list of default themes, these can be cycled through using `<SPC> T n`.

8. **DONE** Toggle status of `ToDo` So this will still work with `<s-Right>` but preferably do it with just `t`

   (a) dealing with fonts you will likely need to fix the font, there is a bug Described here on *Github* that requires the font size listed at approximately line 135 as:

   ```
   dotspacemacs-default-font '("Source Code Pro"
                               :size 13
                               :weight normal
                               :width normal
                               :powerline-scale 1.1)
   ```

   to `13.0` otherwise it doesn't set the font to *point* like it should and the value is entirely ignored, making debugging hard.

   (b) Enabling layers You will need to uncomment lines in order to enable modes like `org` and `markdown`

```lisp
;; Put your css files there
(defvar org-theme-css-dir "~/.emacs.d/org-css/")

(defun toggle-org-custom-inline-style ()
  (interactive)
  (let ((hook 'org-export-before-parsing-hook)
        (fun 'set-org-html-style))
    (if (memq fun (eval hook))
        (progn
          (remove-hook hook fun 'buffer-local)
          (message "Removed %s from %s" (symbol-name fun) (symbol-name
          ↪  hook)))
      (add-hook hook fun nil 'buffer-local)
      (message "Added %s to %s" (symbol-name fun) (symbol-name hook)))))

(defun org-theme ()
  (interactive)
  (let* ((cssdir org-theme-css-dir)
         (css-choices (directory-files cssdir nil ".css$"))
         (css (completing-read "theme: " css-choices nil t)))
    (concat cssdir css)))

(defun set-org-html-style (&optional backend)
  (interactive)
  (when (or (null backend) (eq backend 'html))
 (let ((f (or (and (boundp 'org-theme-css) org-theme-css) (org-theme))))
      (if (file-exists-p f)
          (progn
            (set (make-local-variable 'org-theme-css) f)
            (set (make-local-variable 'org-html-head)
                 (with-temp-buffer
                   (insert "<style type=\"text/css\">\n<!--/*--><![CDAT
                   ↪  A[/*><!--*/\n")
                   (insert-file-contents f)
                   (goto-char (point-max))
                   (insert "\n/*]]>*/-->\n</style>\n")
                   (buffer-string)))
            (set (make-local-variable
            ↪  'org-html-head-include-default-style)
                 nil)
            (message "Set custom style from %s" f))
        (message "Custom header file %s doesnt exist")))))
```

Listing 2: Emacs LISP code to inline custom CSS

Documents I may want to rewrite in `org-mode`

All Stats

Convert and import all R-Work

All NMR

All Abstract Algebra

Tips and Tricks with Org Mode

Folding

Open a second window with `C-x 2` and swap between the windows with `C-x o` (for *O*ther), the other window can be scrolled by using `C-M v`, this is handy for having a second reference window.

The second window can be closed by using `C-x 1` (as in keep **1** window i'm already in)

Working with ToDo/HeadLine/list items

You can add a new `TODO` item with `M-S RET`. Using `M-up/down` a headlines may be moved, they may be demoted with `M-left/right` Using `S-left/right` will toggle to do list items.

Folding

You can fold a headline by using TAB if you are on a headline, the fold level of the entire document may be toggled with `S-TAB`

Trees

Refer to [15]

Key Binding	Description	Description
`SPC m S l`	org-demote-subtree	Move Right
= SPC m S h =	org-promote-subtree	Move Left
= SPC m S k =	org-move-subtree-up	Move UP
= SPC m S j =	org-move-subtree-down	Move Down

[15] Most of this is taken from the *ReadTheDocs* Spacemacs link

Note Taking Guidlines — Workflow

Links

You can create a link by using `C-c C-l` and I think `C-u C-c C-l` will attach an internal link

Exporting Org to Markdown

You should customize org-export-backends and enable the markdown backend. That's: M-x customize-option and then org-export-backends and then arrow down to the checkbox to the left of 'md' and press enter to enable it (or just click on it, if running emacs grqAphically). Then arrow back up and over to 'Apply and Save' (or click on it)

Including LaTeX Fragments — LaTeX

To preview LaTeX documents install `dvipng`, `dvisvgm` or `conver`.

In order to customise the preview customise the variables `org-format-latex-options` and `org-format-latex-hea`

(in order to enable the inline preview use the following press `<s TAB>` to insert a code block.) As hown in (**??**)

```
C-c C-x C-l (org-toggle-latex-fragment)
```

1. Preview is Slow When using `C-c C-x C-l` to preview LaTeX fragments remember that it will generate the fragments using the latex settings corresponding to that buffer, including any additional header arguments you provide.

   Comment out the latex header you are using to give a style sheet, it's going to have lots of crap like TikZ which will slow things right the way down, It will even cripple latex exports, it's better to use that at the very end when you want a nicer product (timing it's about 4.5 X faster)

   Generate latex previews in temporary buffers, that will speed things up a lot

   Try to use `dvi2png` because it is alittle faster, avoid using `imagemagick` because it is slower.[16] Review the variable org-latex-create-formula-image-program in order to verify which is being used.

   Anoter thing that can help is to make sure that you are using `pdflatex` NOT `xelatex`, I timed it on an arbitrary buffer and it went down from 45 to 35 seconds.

[16] macos - Why does org-preview-latex-fragment work about 10 times slower in OSX than on linux? - Stack Overflow

Example LaTeX

$$\sum_{n=0}^{n-1} [ar^n] = a \cdot \frac{1-r^n}{1-r} \tag{9}$$

$$\implies \sum_{n=0}^{\infty} [ar^n] = a \cdot \frac{1-\lim_{n\to\infty}[r^n]}{1-r}$$

$$|r| < 1 \iff \sum_{n=0}^{\infty} [ar^n] = \frac{1}{1-r} \tag{10}$$

DONE Formatting Tables Tables

Let's make a table of key bindings, first we need to insert a table, we'll do this wit a tamplate by typing in `SPC-m-t-n` for *M*ajor mode command, *T*able, *N*ew.

Description for Tables	Key Binding	Notes
Move to next field	`SPC m t l`	.
Move along Table Cells (only for =table.el)	`TAB` and `s-TAB`	There are other bindings
Convert table between `org-mode=/=table.el`	`SPC m t c`	This only works well if you tab between cel
Export to a file	`SPC m t E`	
Plot using *GnuPlot*	`SPC m t p`	`sudo apt install gnuplot`
Toggle Numbers	`SPC m t t o`	think *T*oggle *O*rg Coordinates
Move Table Rows	`SPC m t H/J/K/L`	Just use capital directions
Including LaTeX should work	$\oint z\,\mathrm{d}\,x = 1/_2 \cdot z^2 + C$	

- Why can't I export this as a table?
  - You have to be super careful with the formatting, the table MUST have bar lines on the sides, it doesn't need top and bottom `\hrule` indicators though.
  - The conversion sometimes fucks up and this is where the problem comes in, make sure to use `TABS` to fill in data and when things go wrong just use Block mode to put | on the sides of the converted `org-mode` table, for some reason the conversion puts + there.
- Can my tables include $LaTeX$
  - yeah if it stops working just delete the LaTeX and start again, it's reall finnicky
- Why does the `org-mode` table not include the mark up upon export?
- Why does the `org-mode` table not respect the HTML CSS upon export when the `table.el` clearly does?
-

Multi-Column Cells

If you want to merge cells etc. that's also doable, but it's real finnicky [17] and it doesn't work in *BeOrg*.

Region	Sales							
	Q1		Q2		Q3		Q4	
	foo	bar	foo	bar	foo	bar	foo	bar
North	350	46	253	34	234	42	382	68
South	462	84	511	78	435	45	534	89

Drawers Drawers

Insert a drawer with `SPC m a / , D`.

- Basically this will appear as ordinary text but with the capacity to fold away in emacs
  - So when you perform Visibility Cycling The drawer will not unfold unless you *pull it out* with `TAB`

$$\int 4x^3 dx$$

DONE How to include links to files

DONE Internal links

1. Anchors Anchors can be created by using `«TargetLandingText»` and then accessed by using `[[TargetLandingText][Display Text]]`.

DONE External Files

External files can be linked to by using the syntax `[[./myfile.ext]]` and they can be jumped to with `Enter`, which would be `gf` / `gx` in *Vim*.

DONE Can I tag a specific lines

OK so i have all these notes in here now, like for example multi-column tables, can I tag the specific line or subtree so that I can just jump straight to it??

This would be good for things like analysis where I might need a tag that says, say for example, analysis/real/sequence.

To toggle inline images use `C-c C-x C-v` and to insert them just insert them as a link using `C-c C-l`.

Dragging and Dropping does not work.

[17] Refer to theMailing List for `org-mode`

References

../../Dropbox/Studies/Papers/references