as of 12 MAY 2022

```
This is block with 3 back ticks: green!


This is block with 3 back ticks AND vimdoc:  boring!


Patience !   Takes a few minutes to finish.
shell 137 = out of memory

SOURCE FILE, for vimdoc:
~/code/jimHelp/source/jimHelp.md

CREATE:
jimHelp.txt in ~/code/jimHelp/doc/

PWD:
MUST be ~/code/jimHelp/

PANDOC:
!pandoc --metadata=project:xxx --lua-filter doc/panvimdoc/scripts/skip-blocks.lua --lua-filter doc/panvimdoc/s


FINALLY,
:helptags ALL

# ==========================================
PURPOSE:    Render .md, .txt. .R, .Rmd files
USING:      pandoc, latex, knitr ....
Jump to this file:      |jim_knitr_pandoc_latex|
# ==========================================

as of \today:
    *  To mix latex and .md, must go with pdf, either pandoc or knit
    *  Add r, knitr code to YAML?   then must render as .RMD file
    *  I do not know how to embed latex, produce html or md (github flavor).
```

PDF

```
 PDF   [ignores html, css; also ignores YAML header (pandoc & ::render()]

 To create pdf, just about everything works:  pandoc, markdown, latex, knitr..

  NOTE:   Missing latex .sty ?
        With .tex file, run (in R) tinytex:latemk(*.tex) to install

  fonts installed?  fc-list : family
  (Oct 2021) Can not figure out how to use another font in pandoc:  mainfont:
  is not working.

  (Jan 2022) **Missing font, package?   TinyTex**
  *  update R
  *  keep_tex:  true (in YAML)
  *  at R console  tinytex::lualatex(".... .tex"), or tinytex::latexmk("*.tex")
  *  ~~ some times works, sometimes not ~~

  Lua in *.tex file
  *  see ~/code/publish_project/TEX/
```

```
!pandoc % -f markdown -o %.pdf

!pandoc % -f markdown  -t latex -H ../chapter_break.tex -V linkcolor:blue -V fontsize=11pt -V geometry:margin=
!pandoc % -f markdown  --pdf-engine lualatex -H chapter_break.tex -V linkcolor:blue -V fontsize=11pt -V geomet

!pandoc --metadata=project:JIM --lua-filter doc/panvimdoc/scripts/skip-blocks.lua --lua-filter doc/panvimdoc/s
```

HTML

```
  HTML [to produce HTML with pandoc, all latex is IGNORED.]

I do **not** know how to create fancy HTML files from knitr, pandoc.

 HTML is pain in ass and HUGE time waste.  Pandoc can handle markdown and
 small amounts of latex (math) b/c ppl have added filters or other widgets to
 pandoc.

 Avoid experiments:   will waste time.
 If using Latex, its packages, diagrams with Latex ... must go with PDF.


  *  !pandoc % -f markdown -V linkcolor:blue -V fontsize=11pt -V geometry:margin=0.3in -o out/out.html


-H header
-V or --variable
--pdf-engine=xelatex


*Create pdf from straight txt*
(do not process any markdown)

pandoc balks at processing straight text if it things it sees markdown.
If lucky, !pandoc % -o file.pdf will work.

**BEST**  print_me.sh *.txt file;  then use browser to print and save as .pdf
*.R  - NOPE, Firefox chokes.
```

R, DEFINTIONS, TERSE EXAMPLES

```
Not in this document.
See ~/code/try_things_here/BASE/
```

GIST, GITHUB download to R

```
download.file("https://gist.githubusercontent.com/jimrothstein/c5e148c9a766ab1a1a91464517a0fe1f/raw/61655207c7
              destfile="junk.txt")
readLines(con="junk.txt")
```

neovim, nvim, vim update to latest version

```
{

##  Tue  02 Nov 2021 (also 30 DEC 2021)
-  download nvim.appimage | place in ~/bin/ | will overwrite prior
```

```
-  change permissions to  764
-  do not touch soft link nvim --> nvim.appimage
-  nothing more than this.

##  Wed  09 Feb 2022

-   neovim TERMINAL BUFFER has 2 modes:  Normal (move around as usual, gf, y
etc) and a NEW MODE:  Terminal mode.  This mode means we see BASH cursor.
Anything entered goes there.  There is NO INSERT/EDIT Mode.  You deal with
Terminal mode at the ACTIVE line only.
See #75 Vimcast

-   This mapping copies line , inserts into terminal buffer and runs
noremap <leader>tl Vy<C-w>wpa<CR><C-\><C-n><C-w>pj

}
```

VIM writing_notes

```
*jim_writing_notes1*

http://www.terminally-incoherent.com/blog/2013/06/17/using-vim-for-writing-prose/
:h help-writing
## hard wrap is friend

a=automatic reformat
t=wrap at textwidth

setlocal formatoptions=ant
setlocal textwidth=80
setlocal wrapmargin=0
setlocal foldcolumn=3       "trick, to set left margin


Long parapgarapja l;akdsjf asalkfjas d; asdfk;ladsjf  lk;adjf a;lkaf as;l
asdfjl; adsfl;kj d;as fasdj;lkj afds;lkj

##  Folds
26FEB2022  set to use treesitter;  don't seem to work

##  Turn off indents

(no c indents)

setlocal noautoindent
setlocal nocindent
setlocal nosmartindent
setlocal indentexpr=
```

GIT commands

- Change git push from https to ssh

```
git remote -v   shows using https:
git remote set-url origin git@github.com:jimrothstein/REPOSITORY.git
git remote -v # shows using git:
```

LUA

```
In lua, nil or false evaluate to:  false
0 or '', evaluate to: true


~/code/lua_project/

Lua + neovim:
    *  code is lua.
    *  but calls the neovim API | look careful, can see the vim
        *  api.nvim...command("enew") -- creates new file and edits.
        *  vim.bo[0],buftype=nofile

--  These are vim api ,  called by lua

-- [[  multi-
-- line
-- comments
-- ]]

-- shortcuts:
local cmd = vim.cmd
cmd("pwd")   -- execute vim Ex: command

-- current file name:
:lua print(vim.fn.expand('%')

--  set vim options
:lua vim.api.nvim_command('set nonumber')
:lua vim.api.nvim_command('set number!')        -- toggle
:lua vim.api.nvim_command('echo "Hello, Nvim!"')

--  list buffers, vim.cmd is alias for vim.api.nvim_exec()
:lua vim.cmd('buffers')

-- print
:lua print(_VERSION)
:lua print("hi")

-- print, datatypes
-- Data types are converted correctly
print(vim.api.nvim_eval('1 + 1')) -- 2
print(vim.inspect(vim.api.nvim_eval('[1, 2, 3]'))) -- { 1, 2, 3 }
print(vim.inspect(vim.api.nvim_eval('{"foo": "bar", "baz": "qux"}'))) -- { baz = "qux", foo = "bar" }
print(vim.api.nvim_eval('v:true')) -- true
print(vim.api.nvim_eval('v:null')) -- nil


vim.api.nvim_command('new')

-- To run a lua file
:luafile %
x = 41
if x > 40 then
  print('over 40')
else
  print('under')
end
```

```
-- verb (in init.vim)
-- y{motion} will highlight for you!
-- :au TextYankPost * silent! lua vim.highlight.on_yank()

-- This is a .lua file
-- To source it from .vim:   :luafile <file>
-- :luafile % will also work.
x = "hello"
print(x)

-- tools.lua
local api = vim.api
local M = {}
function M.makeScratch()
  api.nvim.command('enew') -- equal to :enew
  vim.bo[0].buftype=md
end
return M


-- in vim
-- create new command (fails)
-- :command! Scratch lua require'0001_tools'.makeScratch()
--



-- :lua vim.wo.number = true
-- vim.api.nvim_set_win_option('number', true)
-- lua print(vim.wo.number)

-- in a lua file only need following  (and reload)
vim.wo.number = true
vim.wo.number = false
vim.bo.shiftwidth = 4
```

VIMDOC

SOURCE md file

The SOURCE markdown file is located in jimHelp/source. Edit .md file; not the resulting .txt file. Edit .md file; not the resulting .txt file

Resulting txt file.

The resulting txt file will be located in jimHelp/doc

Vim Notes

HELPTAGS and Ctags are NOT related (do not confuse).

for ctags: :h tags-file-format

To change file: edit this file as regular file. Dislike Highighting? :set syntax=off Add a tag: surround new tag with * ; plus prose to describe tag Add a hotlink: ONLY in same file (I think) surround new tag with |

Run :helptags ALL to regenerate file called tags /doc file (singular) : should see this .txt file and tags file

Following sets things up: * open .R file * start R (should be bottom) * :vert h (open help on right)

VIM help 1

```
:h windows.txt
:h vert
:h splitright


:h new   " open new WINDOW
:h enew  " new buffer, in current window

*jim_system_stuff*
:view $VIMRUNTIME
:view $TEMPLATES

*jim_auto_commands*
:h autocmd
:h au



[all docs files](~/docs/)
[code files](~/code/)

:h abbreviation
:h help-summary
:h helphelp
:h help.txt
:h helpgrep
:h usr_toc.txt
:h index

:h startup
:h cmdline
:h exe    (use cmd line to run normal cmds?)
:h startinsert


####  Help for common tasks

:h :abbreviate
:h :augroup
:h :changes
:h :highlight
:h :syntax
:h :command
:h :file
:h :filetype
:h :messages
:h :options  :h options.txt  :h :set
:h :omni
:h :complete  "NOTE:  nvim does NOT have cmd-line completion like C-N, C-P
:h map-listing

:h :scriptnames
:h man  (use vim for manpages)
```

```
*jim_split*
:h :split
:vert help    " open help in vertical split

(N) !!date, insert date

:resize -3 <CR>  " reduce size of window
:vertical resize -3 <CR>

$VIMRUNTIME (inside the image app)
:!ls $VIMRUNTIME

## Windows, splits
:h usr_07.txt
:h usr_08.txt
:h windows.txt
:h CTRL-W


## statusline  %m (modify?) %y (filetype) ...
:h statusline
:echo expand("%m")
:set statusline=%t
:set statusline+=%{&ff}

Ranges (in file)
:h range
:., 'a
:., +2
3 lines below to end - 5 lines
:.+3, $-5

## insert mode
:h insert.txt
:h insert-index
:h i_CTRL-R

<C-R>% inserts file name:
/home/jim/docs/misc_files/005_tech_notes.md

<C-R>=system("ls")  inserts listing


Insert in bulk:
:i or :a  followed by . when done


## Registers
:echo @a
:let @a="hello"


## Plugins
:h Vimux
:call VimuxRunCommand("ls")
:VimuxPromptCommand<CR>

To Close:
:VimuxCloseRunner<CR>
```

```
## Syntax Highlighting
:h usr_06.txt

## vim initialize
:vert h nvim_R
:tab help


## vim help 2
:vert h nvim-R  " opens help to right
:let R_nvimpager = "vertical" default, (can be "tab", "tabnew")

## vim & grep (search both *.R and *.Rmd - note | is escaped)
:grep -EHRn 'binomial' ~/code/**/*.(R\|Rmd)

## vim tabs
tabs   :tabn :tabp :tabnew

READ: cmds to open windows at various localations:  bo, above ...


:h reference_toc
:h help
:h help-summary
:h cmd   (:h ls)
:helpgrep fold*  (no quotes)

"all tags
:h quickref.txt

"index
:h usr_toc.txt

:h reference_toc   (all *.txt files)
:h local-additions (plugins)

:h motions.txt (jumps, motions, find next } etc)


## search
    /foo/+1    find foo  and move +1 line down
/foo/0     find .... but move to beginning of line
/foo/e-1    find ... then move back 1 character.
```

VIM HELP 3 (context)

```
:h i_{}      (insert, delete, visual, ...)

:h :ex_cmd

:h 'option'

:h func()

:h /[     (escape regex character)

:h ft-r-indent    (for plugin r)
```

```
:h ft-json-....   (for plugin json)
```

Pandoc Notes

```
as of \today:
    *  To mix latex and .md, must go with pdf, either pandoc or knit
    *  Add r, knitr code to YAML?   then must render as .RMD file
    *  I do not know how to embed latex, produce html or md (github flavor).

 ===
  PDF   [ignores html, css; also ignores YAML header (pandoc & ::render()]
 ===

  *  Pandoc creates a .tex file (from .md source).   This .tex file is run
      through engine (pdflatex, xelatex ....) to actually output the pdf.

  NOTE:   .tex uses a .sty which I do not have.  USE  knitr:: (with TinyTex
  to locate and install that .sty file)


!pandoc % -f markdown -o %.pdf

!pandoc % -f markdown  -t latex -H ../chapter_break.tex -V linkcolor:blue -V fontsize=11pt -V geometry:margin=
!pandoc % -f markdown  -t latex -H ../chapter_break.tex -V linkcolor:blue -V fontsize=11pt -V geometry:margin=
!pandoc % -f markdown  --pdf-engine xelatex -H chapter_break.tex -V linkcolor:blue -V fontsize=11pt -V geometr

 ====
  HTML [ignores latex]
 ====

  *  !pandoc % -f markdown -V linkcolor:blue -V fontsize=11pt -V geometry:margin=0.3in -o out/out.html
```

LINUX/ZSH notes

gf (goto file)

ZSH

```
*jim_Permissions*
u g o   (user group other)



*grep_vs_ls*
*Grep* always finds words that match a pattern and returns file names of
matches.

ls (+ glob) finds filenames that match a pattern.  Very differnt.
(same in vim)

*jim_GLOB_examples*
Mostly of form ls or ll or print -l    and **/*
example:   print -l ~/code/**/*.(R|Rmd)   # any level, return all .R and .Rmd
files

See my zsh GLOG handwritten notes (till typed in here)
```

```
*zle_widgets* (all commands)
Output from zle -al (~403 cmds)
.accept-and-hold
.accept-and-infer-next-history
.accept-and-menu-complete
.accept-line
.accept-line-and-down-history
.accept-search
.argument-base
.auto-suffix-remove
.auto-suffix-retain
.backward-char
.backward-delete-char
.backward-delete-word
.backward-kill-line
.backward-kill-word
.backward-word
.beep
.beginning-of-buffer-or-history
.beginning-of-history
.beginning-of-line
.beginning-of-line-hist
.bracketed-paste
.capitalize-word
.clear-screen
.complete-word
.copy-prev-shell-word
.copy-prev-word
.copy-region-as-kill
.deactivate-region
.delete-char
.delete-char-or-list
.delete-word
.describe-key-briefly
.digit-argument
.down-case-word
.down-history
.down-line
.down-line-or-history
.down-line-or-search
.emacs-backward-word
.emacs-forward-word
.end-of-buffer-or-history
.end-of-history
.end-of-line
.end-of-line-hist
.end-of-list
.exchange-point-and-mark
.execute-last-named-cmd
.execute-named-cmd
.expand-cmd-path
.expand-history
.expand-or-complete
.expand-or-complete-prefix
.expand-word
.forward-char
.forward-word
.get-line
.gosmacs-transpose-chars
```

```
.history-beginning-search-backward
.history-beginning-search-forward
.history-incremental-pattern-search-backward
.history-incremental-pattern-search-forward
.history-incremental-search-backward
.history-incremental-search-forward
.history-search-backward
.history-search-forward
.infer-next-history
.insert-last-word
.kill-buffer
.kill-line
.kill-region
.kill-whole-line
.kill-word
.list-choices
.list-expand
.magic-space
.menu-complete
.menu-expand-or-complete
.neg-argument
.overwrite-mode
.pound-insert
.push-input
.push-line
.push-line-or-edit
.put-replace-selection
.quote-line
.quote-region
.quoted-insert
.read-command
.recursive-edit
.redisplay
.redo
.reset-prompt
.reverse-menu-complete
.run-help
.select-a-blank-word
.select-a-shell-word
.select-a-word
.select-in-blank-word
.select-in-shell-word
.select-in-word
.self-insert
.self-insert-unmeta
.send-break
.set-local-history
.set-mark-command
.spell-word
.split-undo
.transpose-chars
.transpose-words
.undefined-key
.undo
.universal-argument
.up-case-word
.up-history
.up-line
.up-line-or-history
.up-line-or-search
```

.vi-add-eol
.vi-add-next
.vi-backward-blank-word
.vi-backward-blank-word-end
.vi-backward-char
.vi-backward-delete-char
.vi-backward-kill-word
.vi-backward-word
.vi-backward-word-end
.vi-beginning-of-line
.vi-caps-lock-panic
.vi-change
.vi-change-eol
.vi-change-whole-line
.vi-cmd-mode
.vi-delete
.vi-delete-char
.vi-digit-or-beginning-of-line
.vi-down-case
.vi-down-line-or-history
.vi-end-of-line
.vi-fetch-history
.vi-find-next-char
.vi-find-next-char-skip
.vi-find-prev-char
.vi-find-prev-char-skip
.vi-first-non-blank
.vi-forward-blank-word
.vi-forward-blank-word-end
.vi-forward-char
.vi-forward-word
.vi-forward-word-end
.vi-goto-column
.vi-goto-mark
.vi-goto-mark-line
.vi-history-search-backward
.vi-history-search-forward
.vi-indent
.vi-insert
.vi-insert-bol
.vi-join
.vi-kill-eol
.vi-kill-line
.vi-match-bracket
.vi-open-line-above
.vi-open-line-below
.vi-oper-swap-case
.vi-pound-insert
.vi-put-after
.vi-put-before
.vi-quoted-insert
.vi-repeat-change
.vi-repeat-find
.vi-repeat-search
.vi-replace
.vi-replace-chars
.vi-rev-repeat-find
.vi-rev-repeat-search
.vi-set-buffer
.vi-set-mark

```
.vi-substitute
.vi-swap-case
.vi-undo-change
.vi-unindent
.vi-up-case
.vi-up-line-or-history
.vi-yank
.vi-yank-eol
.vi-yank-whole-line
.visual-line-mode
.visual-mode
.what-cursor-position
.where-is
.which-command
.yank
.yank-pop
_bash_complete-word
_bash_list-choices
_complete_debug
_complete_help
_complete_tag
_correct_filename
_correct_word
_expand_alias
_expand_word
_history-complete-newer
_history-complete-older
_list_expansions
_most_recent_file
_next_tags
_read_comp
accept-and-hold
accept-and-infer-next-history
accept-and-menu-complete
accept-line
accept-line-and-down-history
accept-search
argument-base
auto-suffix-remove
auto-suffix-retain
backward-char
backward-delete-char
backward-delete-word
backward-kill-line
backward-kill-word
backward-word
beep
beginning-of-buffer-or-history
beginning-of-history
beginning-of-line
beginning-of-line-hist
bracketed-paste
capitalize-word
clear-screen
complete-word
copy-prev-shell-word
copy-prev-word
copy-region-as-kill
deactivate-region
delete-char
```

```
delete-char-or-list
delete-word
describe-key-briefly
digit-argument
down-case-word
down-history
down-line
down-line-or-history
down-line-or-search
emacs-backward-word
emacs-forward-word
end-of-buffer-or-history
end-of-history
end-of-line
end-of-line-hist
end-of-list
exchange-point-and-mark
execute-last-named-cmd
execute-named-cmd
expand-cmd-path
expand-history
expand-or-complete
expand-or-complete-prefix
expand-word
forward-char
forward-word
get-line
gosmacs-transpose-chars
history-beginning-search-backward
history-beginning-search-forward
history-incremental-pattern-search-backward
history-incremental-pattern-search-forward
history-incremental-search-backward
history-incremental-search-forward
history-search-backward
history-search-forward
infer-next-history
insert-last-word
kill-buffer
kill-line
kill-region
kill-whole-line
kill-word
list-choices
list-expand
magic-space
menu-complete
menu-expand-or-complete
neg-argument
overwrite-mode
pound-insert
push-input
push-line
push-line-or-edit
put-replace-selection
quote-line
quote-region
quoted-insert
read-command
recursive-edit
```

```
redisplay
redo
reset-prompt
reverse-menu-complete
run-help
select-a-blank-word
select-a-shell-word
select-a-word
select-in-blank-word
select-in-shell-word
select-in-word
self-insert
self-insert-unmeta
send-break
set-local-history
set-mark-command
spell-word
split-undo
transpose-chars
transpose-words
undefined-key
undo
universal-argument
up-case-word
up-history
up-line
up-line-or-history
up-line-or-search
vi-add-eol
vi-add-next
vi-backward-blank-word
vi-backward-blank-word-end
vi-backward-char
vi-backward-delete-char
vi-backward-kill-word
vi-backward-word
vi-backward-word-end
vi-beginning-of-line
vi-caps-lock-panic
vi-change
vi-change-eol
vi-change-whole-line
vi-cmd-mode
vi-delete
vi-delete-char
vi-digit-or-beginning-of-line
vi-down-case
vi-down-line-or-history
vi-end-of-line
vi-fetch-history
vi-find-next-char
vi-find-next-char-skip
vi-find-prev-char
vi-find-prev-char-skip
vi-first-non-blank
vi-forward-blank-word
vi-forward-blank-word-end
vi-forward-char
vi-forward-word
vi-forward-word-end
```

```
vi-goto-column
vi-goto-mark
vi-goto-mark-line
vi-history-search-backward
vi-history-search-forward
vi-indent
vi-insert
vi-insert-bol
vi-join
vi-kill-eol
vi-kill-line
vi-match-bracket
vi-open-line-above
vi-open-line-below
vi-oper-swap-case
vi-pound-insert
vi-put-after
vi-put-before
vi-quoted-insert
vi-repeat-change
vi-repeat-find
vi-repeat-search
vi-replace
vi-replace-chars
vi-rev-repeat-find
vi-rev-repeat-search
vi-set-buffer
vi-set-mark
vi-substitute
vi-swap-case
vi-undo-change
vi-unindent
vi-up-case
vi-up-line-or-history
vi-yank
vi-yank-eol
vi-yank-whole-line
visual-line-mode
visual-mode
what-cursor-position
where-is
which-command
yank
yank-pop
zle-line-finish
zle-line-init
```

BINDKEY

```
*bindkey*  # results, all shortcuts

"^A"-"^C" self-insert
"^D" list-choices
"^E"-"^F" self-insert
"^G" list-expand
"^H" vi-backward-delete-char
"^I" expand-or-complete
"^J" accept-line
"^K" self-insert
```

```
"^L" clear-screen
"^M" accept-line
"^N"-"^P" self-insert
"^Q" vi-quoted-insert
"^R" redisplay
"^S"-"^T" self-insert
"^U" vi-kill-line
"^V" vi-quoted-insert
"^W" vi-backward-kill-word
"^X^R" _read_comp
"^X?" _complete_debug
"^XC" _correct_filename
"^Xa" _expand_alias
"^Xc" _correct_word
"^Xd" _list_expansions
"^Xe" _expand_word
"^Xh" _complete_help
"^Xm" _most_recent_file
"^Xn" _next_tags
"^Xt" _complete_tag
"^X~" _bash_list-choices
"^Y" self-insert
"^Z" backward-delete-word
"^[" vi-cmd-mode
"^[," _history-complete-newer
"^[/" _history-complete-older
"^[OA" up-line-or-history
"^[OB" down-line-or-history
"^[OC" vi-forward-char
"^[OD" vi-backward-char
"^[[1~" vi-beginning-of-line
"^[[200~" bracketed-paste
"^[[2~" overwrite-mode
"^[[3~" vi-delete-char
"^[[4~" vi-end-of-line
"^[[A" up-line-or-history
"^[[B" down-line-or-history
"^[[C" vi-forward-char
"^[[D" vi-backward-char
"^[~" _bash_complete-word
"^\\\\"-"~" self-insert
"^?" vi-backward-delete-char
"\M-^@"-"\M-^?" self-insert
```

XFCE4

Shortcuts: https://docs.xfce.org/apps/terminal/start#keyboard_shortcuts HELP: https://docs.xfce.org/apps/terminal/4.12/start

Based on VTE Widget terminal (gnome uses)

ALT-F10 toggle bet min/max (NOPE!)

ALT-TAB rotate through open windows?

```
##  Thu  19 Nov 2020  Acer Batttery
  *  ACER CB3-431-C7EX
  *  From back (tiny print on labels)
    *  SNID   8120 1450072
    *  SN NXGC7AA001812038A47200
    *  ACER CB-431 Model N16P1
```

Do you sell new battery for this ACER laptop?
CB3-431-C7EX (manuf 3/22/18)
SNID: 81201450072

REST RESTful

*   originally URL linked to file or webpage.
*   more recenty, URI links to payload, HTML/JSON/XML
*   RESTFUL provides stateless operations, architecture (vs SOAP, or others)
* VERBS include GET/POST/ etc etc
Stateless means server keeps no session information.   Each call to server is independent. Examples include HTTP, IP, REST.   But TCP is not stateless.

epub, Calibre, iPad, iCloud, eReader, pdf
-   Claim: iPad does not support Calibre; free Readers for iPad, everyone has fav.
        No, no, no.   Download Calibre software for osx to iPad.   What does not
        work is connecting iPad to Calibre on Laptop.
-   Goodreader for pdf ($20?) - many say best iPad reader.?
- Marvin - no pdf support, but excellent otherwise?

KNITR

knitr -> R & rmarkdown -> Bookdown (~2016) -> Blogdown -> netlify (Hugo, static)
HUGO: md -> html
BOOKDOWN: Rmd   ->html (skips md)

lua is a lightweight language acts like "glue" ; embeds within code; useful in textdoc .

renv   Why I think I do not need (and do not want). Re-creates tidyverse code INSIDE each project, ie local copy of everything inside package. Then takes snapshots as either your code or the any of like libraries changes.   Nice purpose: easily re-create complete environment.   But much too much overhead for my needs! (at this time.)

X11

-   XFCE - many distros, suite of apps, use GTK+ toolkit
-   -   DESKTOP Mgr=Xfdesktop (colors, images, wallpaper)
-   -   FILE Mgr=Thunar (GTK+ toolkit)
        -   others: nautilus
-   -   Windows mgr=xfwm4 (max, min, focus, tiling ...)
-   -   Settings mgr=xfce4-settings-manager (appearance, style, keyboard, ...)
-   -   Terminal=xfce4-terminal (1 of many possible emulators, code that sits
          inside bash?)
-   DISPLAY MGR (DM) = Begins X, then displays (gui) login screen. Many types of DM.
-   chroot -   Without rebooting, chroot means "change root" ie start new shell, change root diretory (to point to a partition)

-   X uses(?) xlib (old), xcb(newer)
-   ncurses lib -?
-   Wayland - next generation (replace?) for X
-   Stack - X at bottom, GNOME or KDE above, NAUTILUS or panels above
-   man Xorg (good) , I have no ~/.xinitrc
-   Terminal is NOT equal to SHELL (explain?)

```
-   GTK+ - C lib, widgets supports X.   Gnone, Win32, etc use GTK+ tools.
-   graphical login? kdm, gdm, xdm (basic) lightdm, sddm aka Display Mgr
-   REMOVE PLUGIN:  vimwiki - how to get rid  | .vimrc - delete references to plugin
##  13 OCT 2018
-   Working: Ranger, newsbeut , updated to Ubuntu 18.04LTS
-   TERMINALS
    -   rxvt, urxvt, terminator, st (not friendly) xfce4-terminal.
```

CURL | YOUTUBE API | GOOGLE API | OAUTH 2.0 |

YOUTUBE (as of 2FEB 2022) API: AIzaSyBIpR4Wee8ZvAiofh41leZvCj7ReVuXRXE SECRET: don't !!

Examples:

cURL write (to standard)

w response after callling example.com

```
curl -w "Response %{response_code}\n" example.com
```

github

curl https://api.github.com/zen

returns lot of kev=value pairs

curl https://api.github.com/users/defunkt

-include headers

curl -i https://api.github.com/users/defunkt

headers only

curl –head

CURL_CONFIG (a FILE)

USAGE curl -K CURL_CONFIG . . .

```
url = example.com
-w "Type:  Hello  %{local_ip} \n"

Misc Notes:
"State"  - cookies used to be used; now state carried in headers
```


```
Misc Notes:
"State"  - cookies used to be used; now state carried in headers
```

Thanks for willing to take a look. Some thoughts to get you and anyone else a start:

- I expect my issue connecting **httr2** and **google api** (youtube) to resolved with simple parameter wrongly set. As usual, it is the journey that is more interesting.
- First, review vignette https://httr2.r-lib.org/articles/wrapping-apis.html, including oauth and github.
- Second, review the command line tool curl https://curl.se.

- Next, look through Google's API documentation + related:
  - https://developers.google.com/youtube/v3/getting-started
  - Try api requests in both Google Oauth2 Playground: https://developers.google.com/oauthplayground/
  - And in Google API Explorer: https://developers.google.com/youtube/v3/docs/

I have done the above without errors. But httr2 code returns 404.

```
404. That's an error.

The requested URL was not found on this server. That's all we know.
```

```
client = oauth_client(id=  client_id,
        token_url  = token_url,
        secret = client_secret,
        key =  API_KEY,
        auth = "body",   # header or body

        name = "youtube_ONE_video_ALL_comments")
```

```
req  <-  request("https://www.googleapis.com/youtube/v3/commentThreads?videoId=Mec9sw1cJk8&part=snippet,replie
req_oauth_auth_code(client = client, auth_url = auth_url, token_params=scope[[1]])
```

```
resp  <- req %>% req_perform()
```

Some Remarks: - Google is but one implementation of various API, oauth technologies. The more you read the more confused you may become (at least for me).
- The R package **gargle** is uses **httr** and therefore not my preference.
- I am using httr2 to automate things; I'd like to understand things using a little as possible: curl, browser and local server running as localhost.
- Most of the R work is done at lower level, such as packages curl and httpuv.