

<3 w/ vi[m]

180976

rules for the session..

#1 thanks for your time!

 this session may go up to 2 hour
 grab your drinks/snacks

#2 no recording plz -

 turn-off the recording
 recording leads less interaction
 no playback - so sharp your focus

#3 ask more questions ; there's no dumb qn

#4 i'll share this slides later

#5 contents/screenshots/memes

 created by self
 fully copyleft; feel free to share/reuse



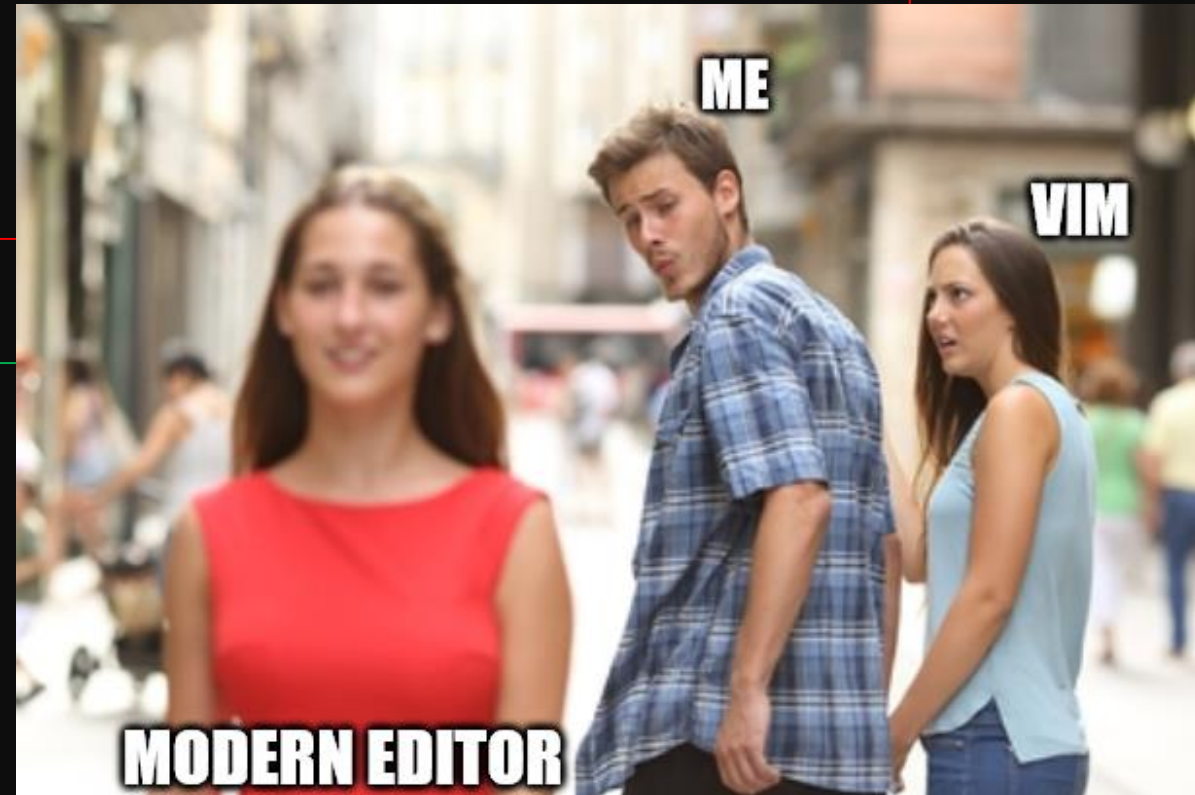
myth vs truth

hey kesavan, stop your nonsense - why i've to love this old vim ?

- i've 21st century modern editors - with GUI - graphics
- vim is old ; vim is so bore ; vim is complex
- quitting vim is impossible
- vim is `colorless`
- vim users are hackers
- vim triggers suicide thinking;

you should **love vim**, because

- ✓ vim cares you
- ✓ vim makes your life better
- ✓ vim is so powerful
- ✓ vim is solid; elder to GNU/Linux
- ✓ vim is `colorful`
- ✓ vim makes you a better dev &
- ✓ to become a hacker, you must be a dev



designed by myself! License : copyleft
(C) Kesavan Muthuvel

checklist - my assumption

☐ vim8

- ✓ on your **TUI** **terminal user interface** - GNU/Linux
- ✓ on your **GUI** **graphical user interface** - Windows

☐ basic OS skill

- ✓ familiar with novice commands: *cd ls cp mv rm* on TUI
- ✓ file navigation & management over explorer on GUI

☐ That's it;

- ☐ You're admitted!

vim is everywhere!

yes,

- ✓ vim is on every kitchen
- ✓ and on every GNU/Linux distro
- ✓ BSD's, Solaris, *nix or whatever most server runs on...

to start w/ vim

- ✓ go to your ~~kitchen~~ \$home terminal
- ✓ command `vim`



powerful vim!



welcome onboard ... seat belt please !

#1 terminal lovers are [cr|l]azy ; so be crazy & lazy .. few commands/terminology

cat	-	concatenate
ls	-	list
mark	-	bookmark

#2 vim has 3 modes or 2 modes or I-don't-know honestly ; but

- | | | |
|----|---|------------------------|
| a) | always ESC ape to get into normal | - file is read only |
| b) | key in i/a/o for insert/append/open | - file ready for write |
| c) | :command from normal | - follow the command |

#3 if vim is not listening you, enforce it !

- ! - don't ask me any question
- ! - don't report any error/warning
- ! - just (do-or-die)

examples

w! - force write
bd! - force buffer delete
q! - quit vim, no questions

#4 " - is for commenting in vim world

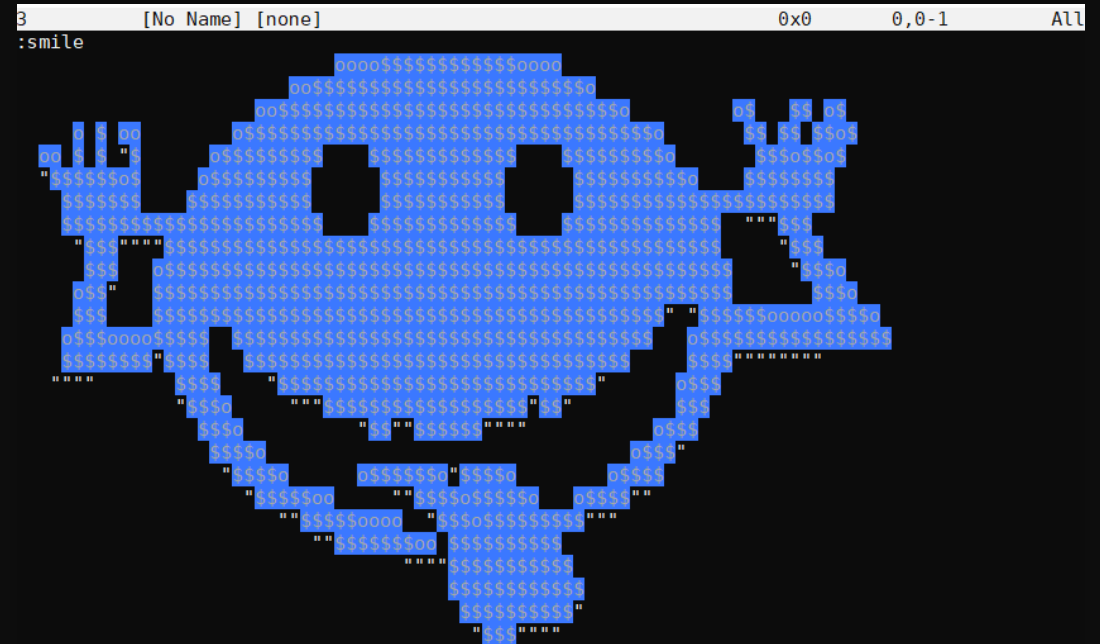
:e - (re)load / (e)dit - in case if you need to reload

:part #1 "basics

1. smile always
2. go to ground
3. play the game
4. guard your data
5. enjoy the break

start w/ smile 😊

:smile "makes you happy



go to the ground: ~/.vimrc

my 1st config

```
vim secret ...sshH!
```

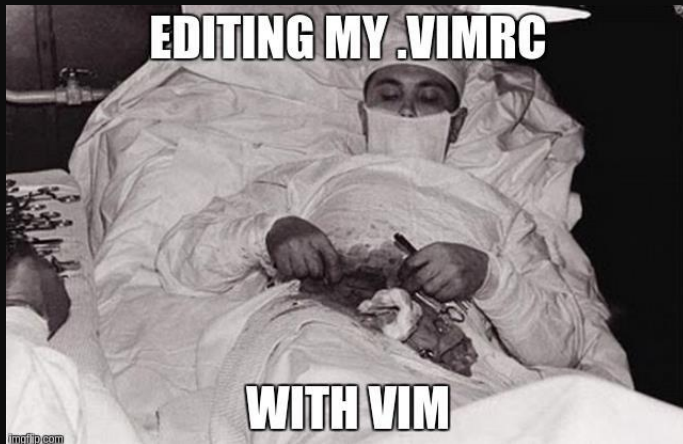
lets have some fun :)

Has own handbook - **.vimrc** - vim runtime config

- train your vim ; tailor your vim
- hack it! break it! fix it!

generally located on home dir :

- ~/ GNU/Linux, BSDs, Solaris
- C:\Users/<xxx>/ MS Windows



regular way:

```
$vi ~/.vimrc
```

```
set number          "set line numbers
```

lazy way:

```
$vi ~/.vimrc
```

```
se nu                "set line numbers
```

key key(s)


i	" into insert mode	#1
	" make changes	#2
ESC	" to normal mode	#3
:w	" write to file	#4
:q	" quit the file	#5

!! Bonus !!

```
:source % " load current file(%) as config
```

```
:wq      " write & quit , remember to be lazy
```

play the game ~/ .vimrc



```
"key stuff
set number           " set line numbers
set nocompatible     " use vim defaults (much better!)
set history=500       " save history
set bs=2             " allow backspace
set laststatus=2     " show status line in 2 last rows
syntax on            " turn on syntax
set directory=~/.vim/tmp " tell vim where's the tmp dir
set undodir=~/.vim/undo-dir " and also history of changes
set undofile         " track my changes

"search
set hlsearch         " highlight the search results
set incsearch        " incremental the search results
set ignorecase       " ignore case ; include all cases
set smartcase

"file encoding & type def
set enc=utf-8
set fileencoding=utf-8
set fileencodings=ucs-bom,utf8,prc
set paste

filetype on
filetype plugin on
filetype indent on

" stop insert asap
autocmd VimEnter * stopinsert " stop insert mode when enter
set noinsertmod              " normal mode than insert
```

recipe secret
...sshH!

little more fun !

as easy as dosa

- ✓ preheat tawa
- ✓ spread wet mixed flour
- ✓ stretch to near round
- ✓ sprinkle oil
- ✓ pack when golden color

search tips

- / search pattern forward
- ? search pattern backward
- n next occurrence
- N previous occurrence
- # find word under cursor forward
- * find word under cursor backward

they're watching...

people love to watch on other's room. keep your data away from preying eyes.

Want to share secret files over mail to colleagues but not to company ?

soln: always follow ``x`` when comes secret

```
$vi -x my_secret.txt      # from terminal
:X                        " from vim
```

- [de|en]crypt files using vim's own algorithm.
- share the key to your friend; unlock w/ the private key
- **warning:** not a strong algorithm; breakable; fair to use

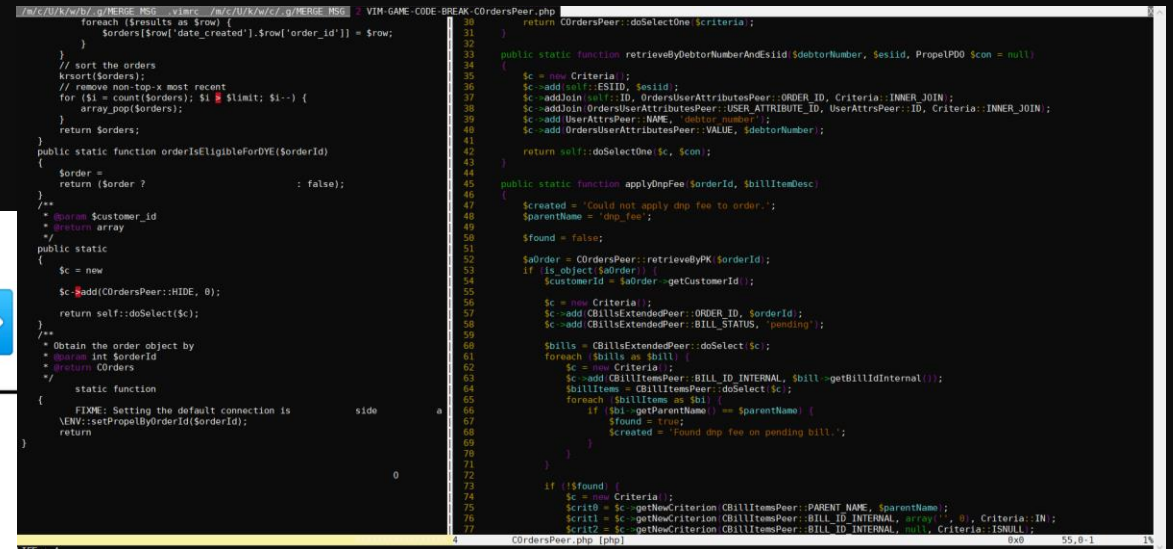
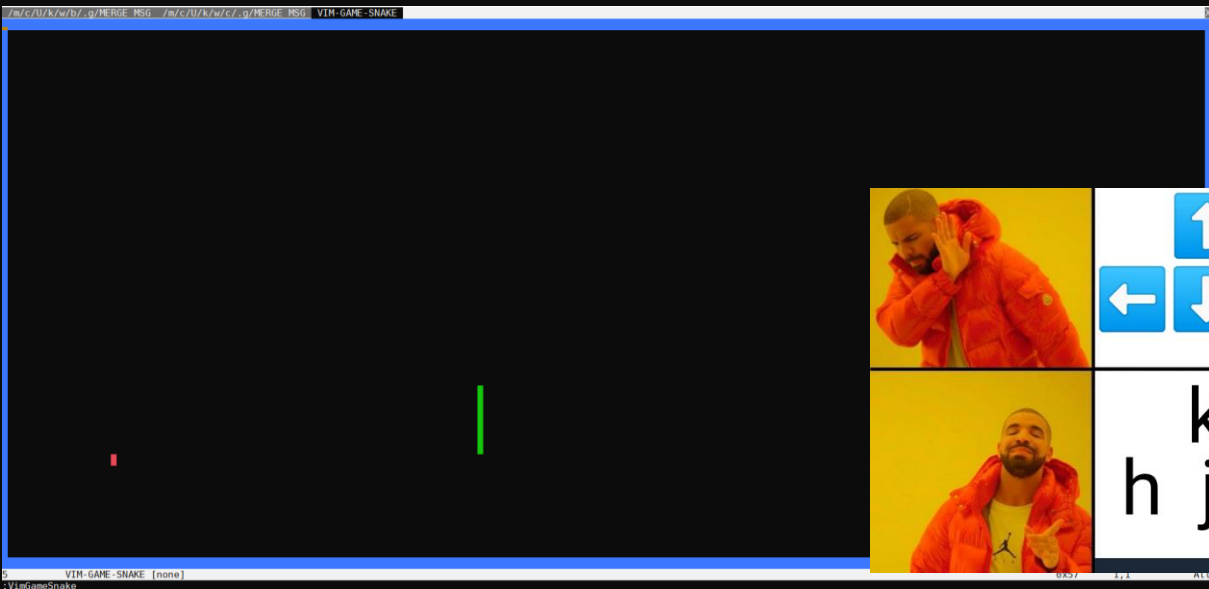
breaktime... or plugin time!

#1 add a **plugin manager** ; below is my fav
download [plug.vim](#) & place under '**autoload**' dir inside [~/vim](#) dir

#2 that's it! restart vim ;

#3 check your plugin manager
:PlugInstall
:VimGame...

```
call plug#begin('~/.vim/plugged')  
    Plug 'johngrib/vim-game-snake'      " 90kids game  
    Plug 'johngrib/vim-game-code-break'  
call plug#end()
```



:part #2 "keys, strikes, xUI, candies

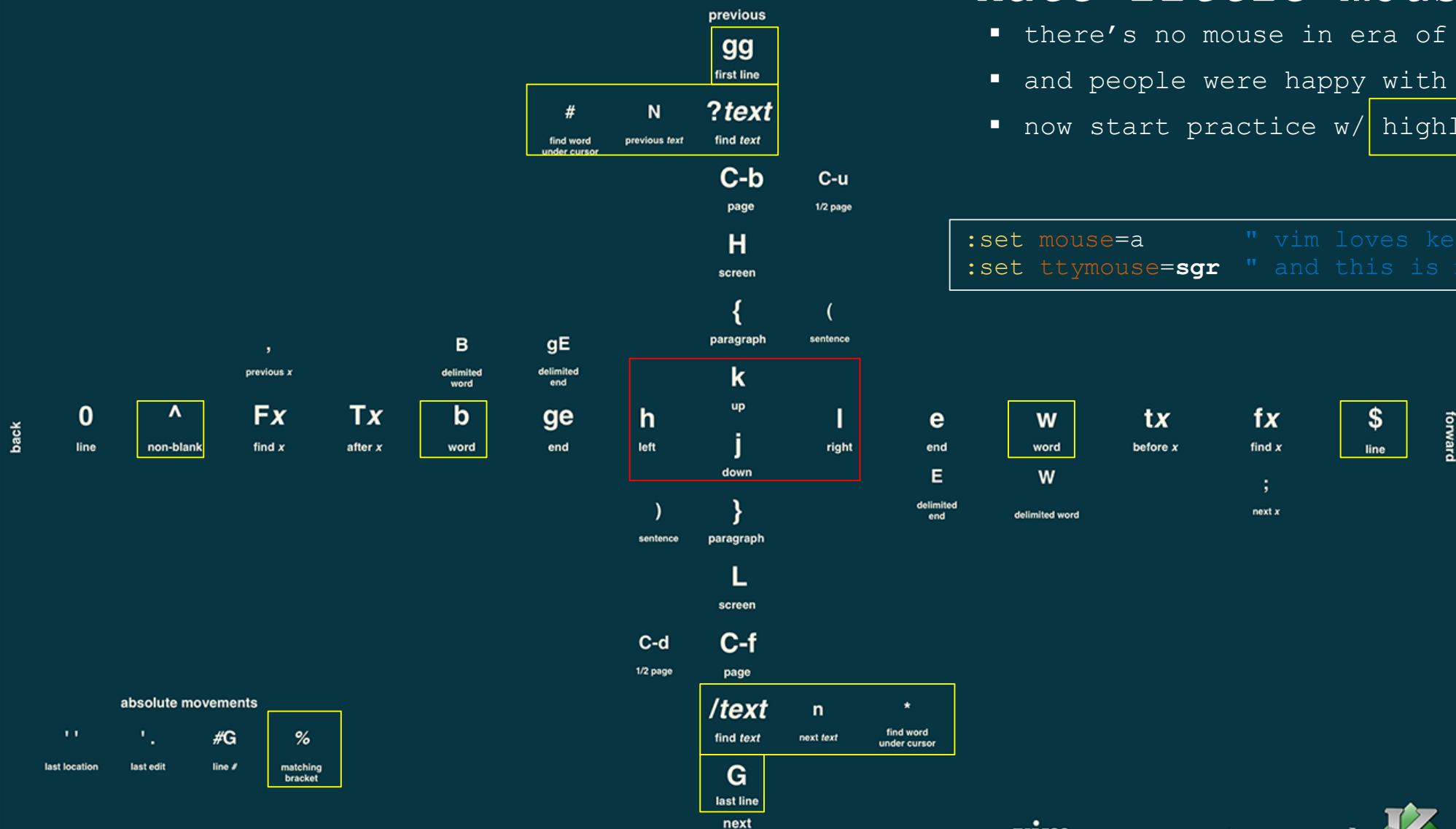
1. little cute mice
2. mark' thy words
3. registers" thy clips
4. color the rainbow
5. terminal vs vim

keyboard or mouse?

vim ♥ keyboard, but never hate little mouse

- there's no mouse in era of vim
- and people were happy with keyboard
- now start practice w/ **highlighted** keys

```
:set mouse=a      " vim loves keyboard, but never mouse
:set ttymouse=sgr " and this is modern mouse
```



vim - movement commands



mark' thy words

always, mark your words - ~ bookmarks

:marks " list all your bookmarks

m[a-zA-Z0-9] " mark a buffer position w/ a char

'x " single quote: jump to line - marked as **x**

`x " backtick: same above but jump to **line + col**

a-z " local marks

A-Z " global marks

Plug 'kshenoy/vim-signature' " enjoy visual marks ☺

jumps - sometime people land nowhere & lost their way

\. " jump to last position

CTRL-o & **CTRL-i** " jump to old/new position

:jumps " show all my way to past

```
61 set directory=~/.vim/tmp
62
63 " stop insert asap
64 autocmd VimEnter * stopinsert
65 set noinsertmode
66
67 "key stuff
68 set number
69 set nocompatible
70 syntax on
71
72 "search
73 set ignorecase
74 set hlsearch
75 set incsearch
76 set ignorecase
77 set smartcase
```

NORMAL > PASTE /home/kesavan/.vim <m utf-8[unix] 48% 73/149 1:1 [18]tra...

:marks

mark	line	col	file/text
'	11	0	"impressing ui
f	54	0	set mouse=a
k	73	0	set ignorecase
u	13	0	set number
H	77	14	~/.bashrc
U	18	15	colorscheme solarized "desert solarized breezy
0	10	0	~/.bashrc
1	26	0	~/.bashrc
2	27	0	~/.bashrc
3	18	43	colorscheme solarized "desert solarized breezy
4	18	43	colorscheme solarized "desert solarized breezy
5	18	43	colorscheme solarized "desert solarized breezy
6	18	43	colorscheme solarized "desert solarized breezy
7	1	0
8	115	0	Plug 'kshenoy/vim-signature'^I"marks +signs
9	115	0	Plug 'kshenoy/vim-signature'^I"marks +signs
"	1	0
[11	0	"impressing ui
]	11	0	"impressing ui
^	117	40	Plug 'RRethy/vim-illuminate'^I"highlight
.	11	0	"impressing ui

Press ENTER or type command to continue.

register" thy clips

Don't forget *register* sweet memories - ~ clips

```
:registers      "list all your clips
:reg           "lazy version

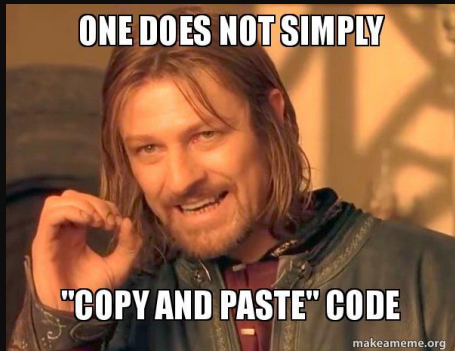
"              " last copied content
0-9            " deleted registers
a-z            " user named registers

"fy            " yank selection into f      NORM MODE
"xp            " paste contents from x     NORM MODE

ctrl-r 5       " paste contents from 5     INS  MODE

                                In VIM world , it's yank & paste

Plug 'junegunn/vim-peekaboo' " enjoy visual registers ☺
```



BONUS

```
Ubuntu 18.04
.vimrc .bashrc > buffers
41 let g:airline#extensions#tagbar#enabled =1
42 tab ball
43
44
45
46
47
48
49
50
51
52
53
54 set mouse=a
55 set ttymouse=sg
56
57 "preserve & restore session
58 "autocmd! VimLeave * mksession!
omatically save the session when leaving Vim
59 "autocmd! VimEnter * source ~/.Session.vim
omatically load the session when entering vim
60 "autocmd VimLeavePre * mksession!
61 set directory=~/.vim/tmp
62
63 " stop insert asap
64 autocmd VimEnter * stopinsert
65 set noinsertmode
66
67 "key stuff
68 set number
69 set nocompatible
e vim defaults (much better!)
70 syntax on
rning on syntax
71
72 "search
73 set ignorecase
74 set hlsearch
75 set incsearch

Special:
": ^[[^[[^[[^[[
-: "

Read-only:
%: /home/kesavan/.vimrc
#: /home/kesavan/.bashrc
/: col
:: marks

Numbered:
0: ^[[^[[^[[^[[
1: ^@
2: Plug 'junegunn/vim-peekabo
3: ^@
4: tmp^@^@" stop insert asap^
5: ^@
6: ^@
7: ^@
8: ^@
9: packadd! dracula^@

Named:
d: shopt -s checkwinsize^@
f: # match all files and zero
i: ^[[^[[

" us
" tu

~/vimrc 48% 73/149 1:1 N... > PASTE [Scratch]
```

extra slide at the appendix on registers.
Yank/paste person? you'll find all your fav
shortcuts

colors & candies – enjoy the rainbow

vim is vibrant & colorful – let's add color to rainbow

curved fonts + colorful theme + right config = beautiful vim

```
set t_Co=256                " support more terminal colors
set guifont=DejaVu\ Sans\ mono:h9
syntax enable
colorscheme desert          " add flavor to your taste
                             " color colo as good!
set cursorline              " where I'm in this island
set colorcolumn=80          " try not to cross the border

" glow in night
Plug 'RRethy/vim-illuminate' " illuminate the variables

" strawberry over vanilla cream
Plug 'vim-airline/vim-airline'
Plug 'vim-airline/vim-airline-themes'

" bring more themes
Plug 'altercation/vim-colors-solarized'
Plug 'fneu/breezy'
Plug 'dracula/vim'
```

tweaks for OS:

\$sudo apt install fonts-powerline

BONUS

```
if exists(":AirlineRefresh")
:AirlineRefresh
endif

let g:airline_theme='breezy'      " gruvbox simple seagull
let g:airline_powerline_fonts = 1
let g:airline_section_b = '%{getcwd()}'

"TABLINE:

let g:airline#extensions#tabline#enabled = 1
let g:airline#extensions#tabline#tabs_label = 'tabs'
let g:airline#extensions#tabline#buffers_label = 'buffers'
let g:airline#extensions#tabline#fnamemod = ':t'
let g:airline#extensions#tabline#show_tab_count = 1
let g:airline#extensions#tabline#show_buffers = 1
let g:airline#extensions#tabline#tab_min_count = 1
let g:airline#extensions#tabline#show_splits = 1
let g:airline#extensions#tabline#show_tab_nr = 1
let g:airline#extensions#tabline#show_tab_type = 1
let g:airline#extensions#tagbar#enabled = 1
```

run the engine from vim

want to execute some command on OS? don't quit vim; there's no escape!

fall into `matrix` ; follow the white rabbit

`:terminal` `:term`
`:vertical bel terminal`

“ open the gateway to other realm
“ vertical terminal on right side bel - below

`sudo apt install sl`
`sudo apt install cmatrix`

BONUS

The screenshot shows a terminal window with the Matrix effect (falling code) and system statistics. The terminal is running on Ubuntu 18.04. The top part of the terminal shows the Matrix effect, which is a series of characters falling from the top of the screen. The bottom part of the terminal shows system statistics, including the number of tasks, CPU usage, memory usage, and a list of running processes.

```
top - 16:50:35 up 3 days, 4:43, 0 users, load average: 0.52, 0.58, 0.59
Tasks: 28 total, 2 running, 26 sleeping, 0 stopped, 0 zombie
%Cpu(s): 28.8 us, 29.8 sy, 0.0 ni, 40.5 id, 0.0 wa, 0.8 hi, 0.0 si, 0.0 st
KiB Mem : 33397884 total, 17471112 free, 15697420 used, 229352 buff/cache
KiB Swap: 62367484 total, 62290196 free, 77288 used, 17566732 avail Mem

PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     TIME+ COMMAND
14762 kesavan  20   0 34832 13876 8368  R   20.3   0.0   0:50.18 vim
14809 kesavan  20   0 14444 1388 1080  R    9.0   0.0   0:13.29 cmatrix
14832 kesavan  20   0 17624 2080 1588  R    1.0   0.0   0:00.48 top
14833 kesavan  20   0 10928 1036  888  S    1.0   0.0   0:00.03 sl
14817 kesavan  20   0 17880 3880 3712  S    0.7   0.0   0:00.23 bash
      1 root      20   0  8304   132  184  S    0.0   0.0   0:01.04 init
3397  root      20   0 367540 16644 16576  S    0.0   0.0   0:05.87 apache2
3400  www-data  20   0 380596 40196 24960  S    0.0   0.1   0:13.19 apache2
3402  www-data  20   0 384704 53660 34052  S    0.0   0.2   1:04.38 apache2
3414  www-data  20   0 378992 47324 33440  S    0.0   0.1   1:08.85 apache2
3606  mysql      20   0 10660   628   596  S    0.0   0.0   0:00.25 mysqld safe
4001  mysql      20   0 2667500 526724 5752  S    0.0   1.6 109:17.68 mysqld
4825  www-data  20   0 377132 38248 26488  S    0.0   0.1   0:19.81 apache2
4851  www-data  20   0 377164 38780 26928  S    0.0   0.1   0:16.71 apache2
4852  www-data  20   0 380956 41464 25852  S    0.0   0.1   0:09.02 apache2
4853  www-data  20   0 378676 41120 27620  S    0.0   0.1   0:41.44 apache2
4861  www-data  20   0 380684 46536 38964  S    0.0   0.1   0:40.07 apache2
```

:part #3 "buffers windows tabs

1. watch the street
2. handle more buffers
3. break the windows
4. fix with tabs
5. save your space

let's watch buf from wintab

- **buffer** - **file in memory** - ready for r/w operation
 - saving buffer is writing into physical file at OS level
 - multiple buffers can be opened, parallely!
- **window** - **view port** to watch buffers
 - a window can be used to look after multi buffers
- **tabpage** - **hold one or more window**
 - collections of windows ;
 - manage the workspace

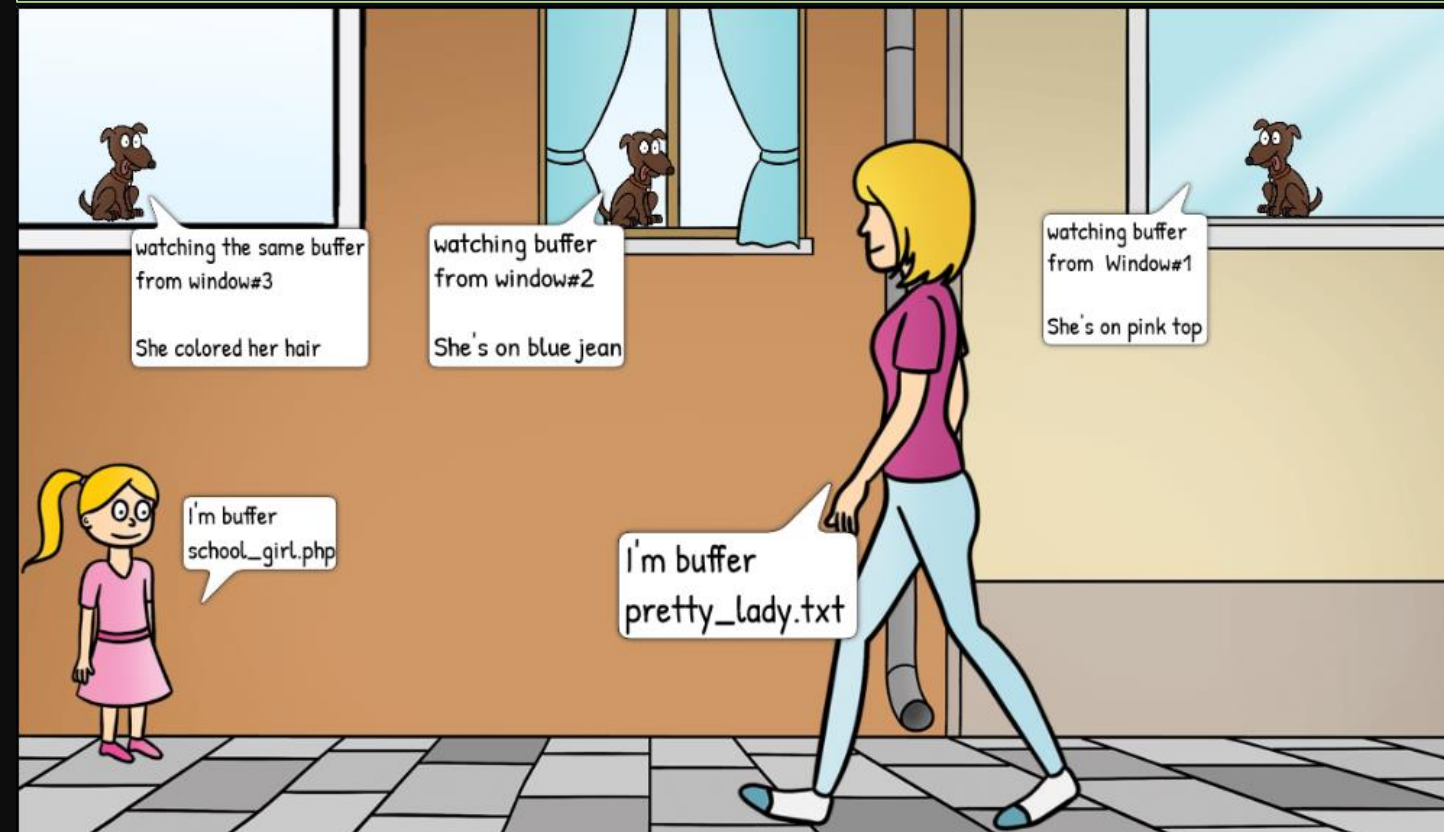
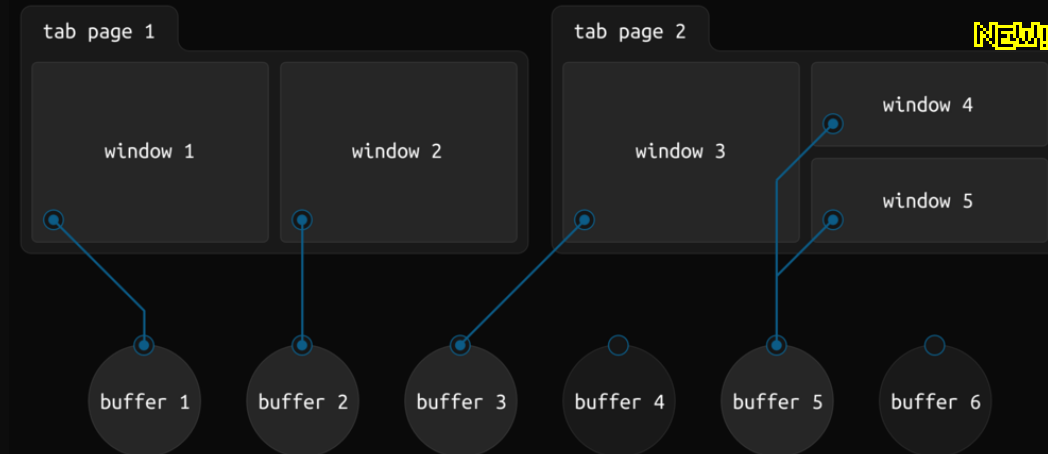


commands to handle multi files operations

- **buffer** - bufdo
- **window** - windo
- **tab** - tabdo

```
:bufdo %s/foo/bar/g
- substitute foo w/ bar all buffers
:windo %s/foo/bar/g
- substitute foo w/ bar on buffers in current window
- tab - tabdo
```

NEW!




handle more w/ **buffers**; yes, you can!

buffer - file in memory - ready for r/w operation

```
$vi abc.txt xyz.php pq.js # handle more files
```

```
$vi * # load all files
```



```
:buffers :ls :files - list all buffers w/ <id>
```

```
:bn[ext] - buffer next
```

```
:bp[revious] - buffer previous
```

```
:bf[irst]/bl[ast] - buffer first/last
```

```
:b xy - buffer xy.php; w/ auto-complete
```


```
:b# - buffer w/ id
```

break the windows

window – just a view port to look over buffer

- closing window doesn't close the buffer
- buffer still loaded on vim

window commands are starts with ctrl-w

c-w s	– s plit window horizontally
c-w v	– split window v ertically
 c-w w	– switch to other window (clockwise)
c-w ><	– adjust the horizontal width
c-w +-	– adjust the vertical height

c-w q	– quit the window
c-w o	– quit all windows, keep this only

:split	:sp
:vsplit	:vs

:quit	q
:only	:on

:windo – execute commands on all open windows

exercise #todo

:belowright vertical terminal
:aboveleft vertical terminal



BONUS

fix w/ tabs

- tab page – collection of windows & buffers
- grouping mech to keep one tab for related files
- tab is more GUI friendly; your mice love tabs

:tab ball " open all buffers in separate tabs*

:tabo " tabs into buffers

:tabs – list all open tabs

:tabnew – open new tab page

:tabnext – tab page next

:tabp[revious] – tab page previous

* **:set tabpagemax=99** "allow more tabs

BONUS

save your workspace

- ✓ keep all your opened tabs & windows layouts
- ✓ reload all previous settings for buffers/vim

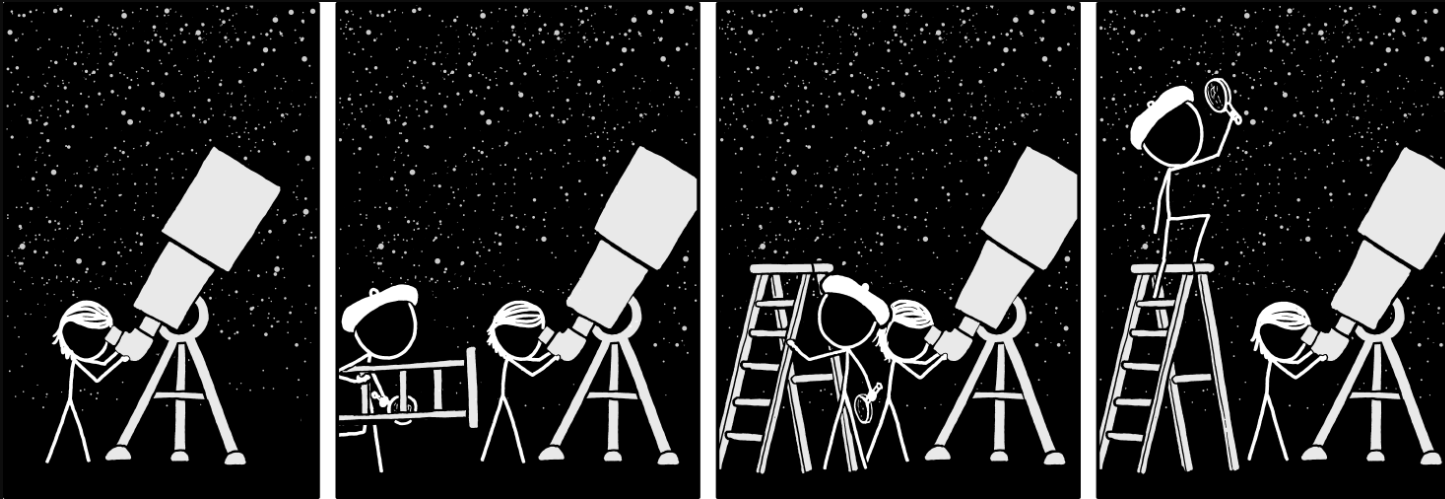
```
:mksession ~/Session.vim      "make session -~/Session.vim  
:source ~/Session.vim        "load session
```



- much useful in GUI; for 2K kids who look for modern experience

```
"preserve & restore session  
autocmd! VimLeave * mksession!           " save the session when leaving  
autocmd! VimEnter * source ~/Session.vim " load the session when entering  
autocmd VimLeavePre * mksession!
```

BONUS



:part #4 "explore, travel, search, repeat


1. explore local ground
2. discover external space
3. fuzz the finder
4. grep goes rip
5. travel against time

xkcd.com/1522 CC

explore the world

handling multiple jobs? busy w/ high demand ?

always explore outside world from vim with:



```
:Ex      - tmp explore & work in existing window
:Sex     - tmp explore & work in new horizontal window
:Vex     - tmp explore(left) & work in vertical window
:Lex     - permanent explorer on Left - modern look

:Tex     - quick mode - tab version
:Rex     - Return to explorer
```

! does reverse when comes with exploring

```
:Ex!     - :Ex in vertical split
:Sex!    - same of :Sex, but vertical
:Vex!    - right :Vex
:Lex!    - keep it Right

:Tex!    - NOT Allowed
:Rex!    - NOT Allowed
```

:help Explore

```
: [N]Explore[!]  [dir]... Explore directory of current file      :Explore
: [N]Hexplore[!] [dir]... Horizontal Split & Explore             :Hexplore
: [N]Lexplore[!] [dir]... Left Explorer Toggle                   :Lexplore
: [N]Sexplore[!] [dir]... Split&Explore current file's directory :Sexplore
: [N]Vexplore[!] [dir]... Vertical Split & Explore                :Vexplore
: Texplore       [dir]... Tab & Explore                           :Texplore
: Rexplore       ... Return to/from Explorer                     :Rexplore

Used with :Explore */pattern : (also see netrw-starstar)
:Nexplore..... go to next matching file                       :Nexplore
:Pexplore..... go to previous matching file                     :Pexplore
```

discover external space

```
1: vimrc 5: canvas.patch 7: file.csv 8: config.ders-cms.php 9:
clean_duplicate_orders.dif 96 endif
cmatrix.log 97
composer-setup.php 98 let g:airline_powerline_fonts = 1
composer.phar* 99 let g:airline_section_b = '%{getcwd()}'
COrdersPeer.php* 100
ctags-5.8.tar.gz 101 let g:airline#extensions#tabline#enable
file.csv 102 let g:airline#extensions#tabline#buffer
GoDaddySecureCertificateAu 103 let g:airline#extensions#tabline#buffer
GoDaddySecureCertificateAu 104 let g:airline#extensions#tabline#fname
GoDaddySecureCertificateAu 105 "let g:airline#extensions#tabline#fname
presenting.vim.md* 106
ql.csv 107
ql.sql 108 "let g:airline#extensions#tabline#buffer
quick.sh* 109 "let g:airline#extensions#tabline#tab
Session-tabs.vim 110 "let g:airline_theme='breezy'
Session.vim 111 "let g:airline#extensions#tabline#fname
Session.vim.6.11 112 "let g:airline#extensions#tabline#show
Session.vim.6.11b 113 "let g:airline#extensions#tabline#show
Session.vim.X 114 "let g:airline#extensions#tabline#tab
Session.vim.X 115 "let g:airline#extensions#tabline#show
sl.log 116 "let g:airline#extensions#tabline#show
vimrc
sp_remove_customer_data.sql 3 use FireEngine\Helper\Text;
startall 4
test.delme 5 /**
tmp.php 6 * Skeleton subclass for performing c
tmp.php.bk 7 *
tmp.php.bk2 8 *
trust.gpg.zip 9 *
views.sql 10 * You should add additional methods
vim-typescript* 11 * application requirements. This cl
12 * long as it does not already exist
13 *

/home/kesavan COrdersPeer.php
NERDTree Menu. Use j/k/enter, or the shortcuts indicated
=====
> (!)Execute file
(a)dd a childnode
(m)ove the current node
(d)elete the current node
(c)opy the current node
print (p)ath to screen
(l)ist the current node
Run (s)ystem command in this directory
```

another better tool to discover more stuff - **NERDTree**

```
Plug 'preservim/nerdtree' " explorer
Plug 'ryanoasis/vim-devicons' " icons
autocmd BufWinEnter * silent NERDTreeMirror " mirror it
```

```
O - open file in new buffer
t - open file in new tab
s - open file vertical split
i - open file horizontal split
```

- ✓ file management made easy
- ✓ multiple files ? no problem
- ✓ love visualization? yes it is!
- ✓ **(B)**ookmark those key files

```
m " menu offers quick file operations!
B " show bookmarks
```

```
:Bookmark " bookmark the file within NerdTree
```



fuzzy find Ctrl-P

Ctrl-P - unlock the MRUs/buffers/files ... index'm all

```
Plug 'ctrlpvim/ctrlp.vim'      " Ctrl-P
```

ctrl-p " open the fuzzy finder 

ctrl-o " open file

ctrl-t " open file in new tab

ctrl-z " (un)mark to open multi files


```
~/ .vimrc
> application/reports/AlbertaNewOrdersPlacedReport.php
> application/reports/UpdatedOrdersPlacedDeReport.php
> application/reports/IqorNewOrdersPlacedReport.php
> application/reports/OrdersPlacedBureauReport.php
> application/reports/OrdersPlacedDeReport.php
mru files buf <->
>>> OrdersPla_
```

ctrl-b or **ctrl-f** - " switch between buffers/mru/files
" mru : **m**ost **r**ecent **u**sed

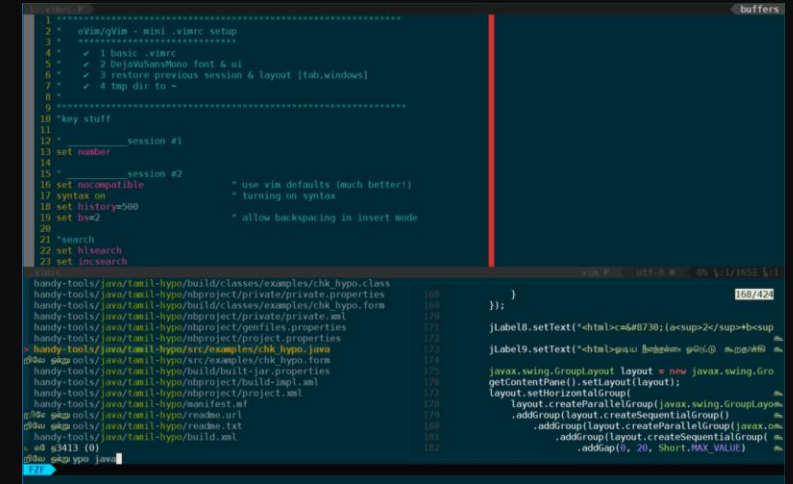



grep vs vimgrep vs rg ... fzf!

Vim has it's own search tool - vimgrep/vimgrepadd

 **/findme** "search & highlight the pattern
/[cC]Order "search & highlight the regex
:% s/idly/dosa/gc "substitute idly with dosa

"search all buffers
:execute "bufdo vimgrep /needle/ %" | copen " search & pop in quick list
:cexpr [] " cleanup quick list
:cclose or :q (from quick-list) " close quick-list*



 Plug 'jeetsukumaran/vim-buffersaurus' " lets grep the needle, easy/lazy
:**Bsgrep** <pattern> " grep the pattern across the buffers


try ripgrep (rg) - faster than grep. Also install bat over cat

\$sudo add-apt-repository ppa:x4121/**ripgrep** # from terminal
\$sudo apt install bat **ripgrep**

BONUS

"get that rocket engine

Plug 'junegunn/fzf.vim' " fzf <3 vim
Plug 'junegunn/fzf',{ 'do': { -> fzf#install() } } " fzf binary

 **:Files** " find for files
:Rg " find in files
:Buffers " list :buffers

travel against time

Vim has built-in time machine , yes!

You can forward to your past or back to future as you want.
All you need is to ensure you've **undodir** & **undofile** set in place

```
:earlier 30s " forward to past 30 sec
:later 1h " back to future 1 hour
:ea 5d " forward to past 5 days
:lat 7f " move to 7th future state
```

s/m/h/d - sec/min/hr/day f - file writes



```
let vimDir = '$HOME/.vim'
let &runtimpath.='','.vimDir
if has('persistent_undo')
    let myUndoDir = expand(vimDir . '/undodir')
    " Create dirs
    call system('mkdir ' . vimDir)
    call system('mkdir ' . myUndoDir)
    let &undodir = myUndoDir
    set undofile
endif
```

```
Plug 'mbbill/undotree' " travel time
:UndotreeShow " reborn in past
```

short form

" time machine - it's already there
set undodir=~/.vim/undo-dir
set undofile

```
* >4< S (32 minutes ago)
* 3 (32 minutes ago)
* 2 (32 minutes ago)
* 1 (32 minutes ago)
* 0 (Original)
```

BONUS

:part #6 `"vim-as-an-ide`

1. repeat some nonsense
2. compare that data
3. edit it vertically
4. view the x-ray
5. vim as debugger

repeat those nonsenses

macro - record this ; play forever

:q{your letter}{series of commands}q

to record: q<alphanum>

to stop : q to quit

apply on lines matching pattern :g/pattern/normal! @a

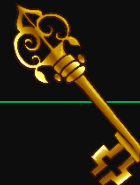
apply on next 100 lines :100@a

apply on from lines 34-65 :43,65normal! @a

apply on all lines :%norm! @a

to record: q<alphanum> actions q

to repeat: :norm! @<alphanum>



```
1:delme.sql+ =
1 INSERT INTO X VALUES ('ABC')
2 INSERT INTO X VALUES ('ABC')
3 INSERT INTO X VALUES ('ABC')
4 INSERT INTO X VALUES ('ABC')
5 INSERT INTO X VALUES ('ABC')
6 INSERT INTO X VALUES ('ABC')
7 INSERT INTO X VALUES ('ABC')
8 INSERT INTO X VALUES ('ABC')
9 INSERT INTO X VALUES ('ABC')
10 INSERT INTO X VALUES ('ABC')
11 INSERT INTO X VALUES ('ABC')
12 INSERT INTO X VALUES ('ABC')
13 INSERT INTO X VALUES ('ABC')
14 INSERT INTO X VALUES ('ABC')
15 INSERT INTO X VALUES ('ABC')
16 INSERT INTO X VALUES ('ABC')
17 INSERT INTO X VALUES ('ABC')
18 INSERT INTO X VALUES ('ABC')
19 INSERT INTO X VALUES ('ABC')
20 INSERT INTO X VALUES ('ABC')
21 INSERT INTO X VALUES ('ABC')
22 INSERT INTO X VALUES ('ABC')
23 INSERT INTO X VALUES ('ABC')
24 INSERT INTO X VALUES ('ABC')
25 INSERT INTO X VALUES ('ABC')
26 INSERT INTO X VALUES ('ABC')
27 INSERT INTO X VALUES ('ABC')
28 INSERT INTO X VALUES ('ABC')
29 INSERT INTO X VALUES ('ABC')
30 INSERT INTO X VALUES ('ABC')
31 INSERT INTO X VALUES ('ABC')
32 INSERT INTO X VALUES ('ABC')
33 INSERT INTO X VALUES ('ABC')
34 INSERT INTO X VALUES ('ABC')
35 INSERT INTO X VALUES ('ABC')
36 INSERT INTO X VALUES ('ABC')
37 INSERT INTO X VALUES ('ABC')
38 INSERT INTO X VALUES ('ABC')
39 INSERT INTO X VALUES ('ABC')
40 INSERT INTO X VALUES ('ABC')
41 INSERT INTO X VALUES ('ABC')
```

sample#1: append ; at EOL

recording - qt\$A;<ESC>q
playing - 2,\$norm! @t

sample#2: delete empty lines

recording - :qx dd <ESC>q
playing - :g/^\$/norm! @x


vimdiff - always compare & decide

looks alike ? compare before choose!

always compare

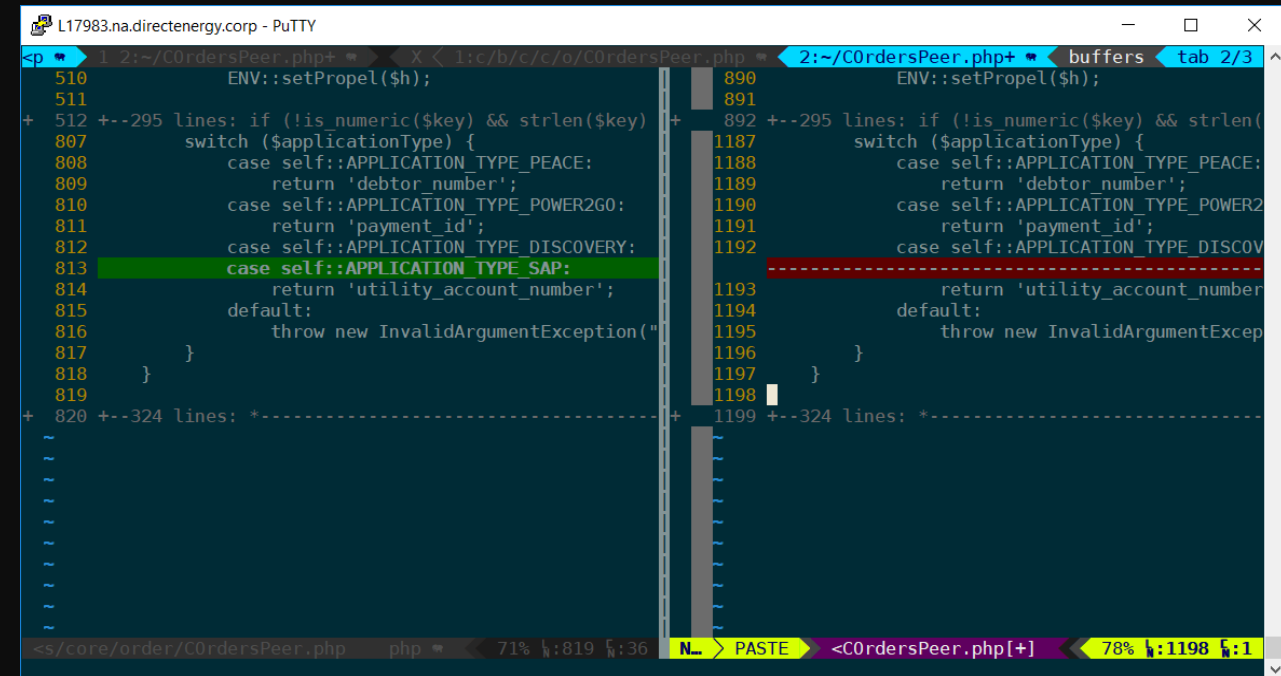
```
$vim -d abc.txt abcd.txt
```

]	c	-	next diff
[c	-	prev diff
do		-	diff obtain
dp		-	diff put
zi		-	toggle fold

 **windo** :diffthis
windo :diffupdate

 :set t_Co=1 " if there's too much color

```
hl DiffAdd      cterm=BOLD ctermfg=NONE ctermbg=22
hl DiffDelete   cterm=BOLD ctermfg=NONE ctermbg=52
hl DiffChange    cterm=BOLD ctermfg=NONE ctermbg=23
hl DiffText      cterm=BOLD ctermfg=NONE ctermbg=23
```



run up/down, not just back/forth

edit your job in vertical (column) mode



ctrl-v - select the columns

I	- I nsert
r	- r eplace
d	- d elete
c	- c hange/edit

on hurry burry? no time to unzip?

vim has xray eyes ; it can play on compressed items. so finish your task in quick way

```
$vim something.zip archive.tar.gz #from terminal
```

- will list archived files
- which you can edit/save without extracting
- supported archives: tar.gz, tgz, zip, etc

vim as debugger

- ❑ **vdebug** can help you debug the code directly from vim
- ❑ but it too early now ; next session
- ❑ and too much plugins === too much of makeups



```
:part #x  "goals, recap, extra bonus
```

free-bonus

```
" buffer navigation - shortcuts
" Ctrl Left/h & Right/l cycle between buffers

let mapleader = "\<Space>"

noremap <silent> <C-left> :bprev<CR>
noremap <silent> <C-h> :bprev!<CR>
noremap <silent> <C-right> :bnext<CR>
noremap <silent> <C-l> :bnext!<CR>
noremap <silent> <S-S> :Files<CR>
```

buffers v2

[d]elete vs [w]ipeout

```
buffer - file in memory - ready for r/w operation
$vi abc.txt xyz.php pq.js " handle more files
$vi * " load all files

:buffers :ls - list all buffers w/ <id>
:bn[ext] - buffer next
:bp[revious] - buffer previous
:bf[irst]/bl[ast] - buffer first/last
:b xy - buffer xy.php;
:b <id> - buffer w/ id
```



dirty recap

```
:bd - deletes the current buffer, error if there are unwritten changes
:bd! - deletes the current buffer, no error if unwritten changes
:bufdo bd - deletes all buffers, stops at first error (unwritten changes)
:bufdo! bd - deletes all buffers except those with unwritten changes
:bufdo! bd! - deletes all buffers, no error on any unwritten changes

:bw - completely deletes the current buffer, error if there are unwritten changes
:bw! - completely deletes the current buffer, no error if unwritten changes
:bufdo bw - completely deletes all buffers, stops at first error (unwritten changes)
:bufdo! bw - completely deletes all buffers except those with unwritten changes
:bufdo! bw! - completely deletes all buffers, no error on any unwritten changes

:set confirm - confirm changes (Yes, No, Cancel) instead of error
:tabe +Nbuf - open buffer number N in new tab
```

register" thy clips w/ detail

	<u>yank</u>	normal/visual	command	action/outcome
To local reg	y		:y	"yank selected into default reg(")
	yy Y		:yy :Y	"yank line into default reg(")
	"a-z0-9y		:ya-z0-9	"yank into named/numbered reg
To OS reg	"*y		:y *+	"yank into OS reg
To remote OS (PuTTY/SSH)	ALT+SHIFT LEFT CLICK			"and select the content "yank into remote OS memory
	SHIFT LEFT CLICK			"and select the content (dirty) "yank into remote OS memory
	:se mouse=c			"set copy mode
	SELECT & left click			"yank into remote OS

	<u>paste</u>	normal/visual	command	action/outcome
From local reg	p P		:put	"paste from default reg; "(p)after / (P)before cursor
	"a-z0-9p		:puta-z0-9	"paste from named/numbered reg
From OS reg	"*+p		:put *+	"paste from OS reg
From remote OS (PuTTY/SSH)	SHIFT RIGHT CLICK			"paste from REMOTE OS yank
COMMAND mode	CTRL-R "			"from reg
	CTRL-R *			"from OS reg

registers & description	
0	last copied clip (yank only)
1-9	numbered regs - FILO algorithm
a-z	named regs - associative array
"	last clip (cut/delete/yank)
*	gnu/mac/win OS memory
+	gnu only special OS memory

INSERT mode - access the registers	
<CTRL+r> 0-9a-z	paste reg content
<CTRL+r> =	quick vim calc'r



pasting on command line :	
<CTRL+r> "	paste default reg
<CTRL+r> =	quick vim calc'r
<CTRL+r> =	quick vim calc'r



" friendly right click menus
Plug 'kvngvikram/rightclick-macros'

" make it as a shortcut
:noremap y "*y

bonus #1

- bring colors & flavors to your life
- i too love black & white but colors are cool

```
$ lolcat --help

Usage: lolcat [OPTION]... [FILE]...

Concatenate FILE(s), or standard input, to standard output.
With no FILE, or when FILE is -, read standard input.

--spread, -p <f>:  Rainbow spread (default: 3.0)
--freq, -F <f>:   Rainbow frequency (default: 0.1)
--seed, -S <i>:   Rainbow seed, 0 = random (default: 0)
--animate, -a:   Enable psychedelics
--duration, -d <i>: Animation duration (default: 12)
--speed, -s <f>:  Animation speed (default: 20.0)
--force, -f:     Force color even when stdout is not a tty
--version, -v:   Print version and exit
--help, -h:     Show this message

Examples:
lolcat f - g      Output f's contents, then standard input, then g's contents.
lolcat            Copy standard input to standard output.
fortune | lolcat  Display a rainbow cookie.

Report lolcat bugs to <http://www.github.org/busyloop/lolcat/issues>
lolcat home page: <http://www.github.org/busyloop/lolcat/>
Report lolcat translation bugs to <http://speaklolcat.com/>

$ fortune | cowsay | lolcat

-----
/ Your own qualities will help prevent \
\ your advancement in the world.      /
-----

      ^__^
      (oo)\_______
      (__)\       )\/\
      ||----w |
      ||     ||

$ █
```

lolcat

#add colors

#enjoy rainbow



bonus – keep an eye on jobs

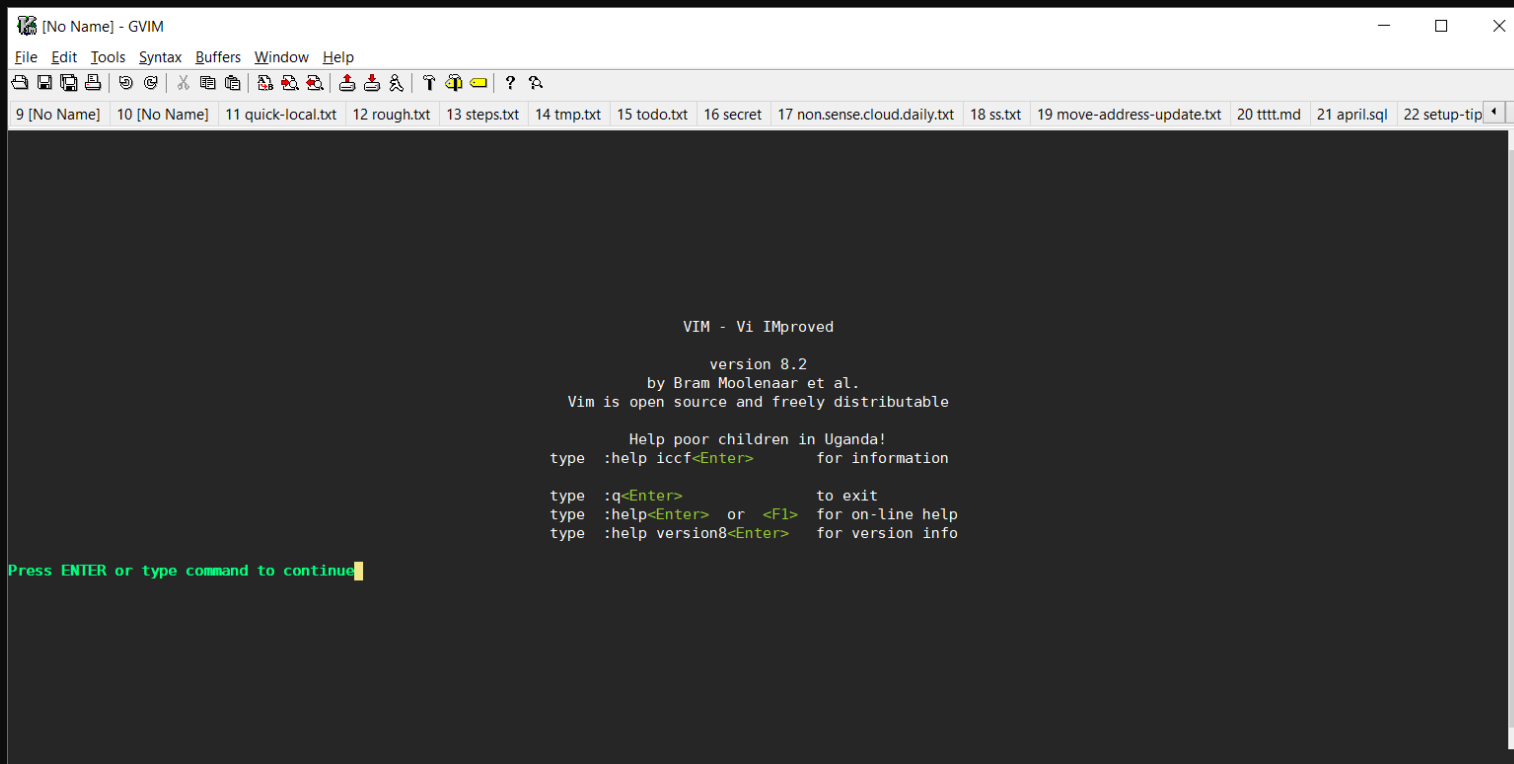
- running multiple jobs/threads ?
- not sure who is the culprit to blame
- kill that right zombie/daemon
- find mother & her (original) daughters

```
ps auxwf | grep -B8 " [2]533 "  
    # display process table in GNU/Linux with  
    # parent/child relationships, searching  
    # for process id 2533
```

gVIM – sister of vim, GUI version

#1 – hope you've GUI gVIM ; start by calling it "gvim -y"

#2 – modern look ; exclusively for **2k** kids



bonus #4



- careful of what you're doing
- **rm** - think twice before execute

#a note in author's history :

*"forget to note down a single space between ~
and / in "rm -rf ~ /.gconf/apps/panel"
:(guess what happens to my data. :(
#omg #linux #helpme"*

@kesavan2000in

bonus #5

enjoy the basic express calc in vim

```
" from insert mode
```

```
■ CTRL-r =sqrt(4)+sin(90)
```

```
" 2.893997
```

```
■ CTRL-k p* roughly equals CTRL-r = 22/7.0
```

```
" π roughly equals 3.142857
```

```
:digraphs
```

```
" list all symbols
```



climax – fork & bomb (do it again)

- time to break?
- have fun

```
:() { :|:& };;:
```

warning #1 – don't inject the above

warning #2 – read warning #1 again

hint: it is actually

```
 () {  |  & } ; 
```

thanks to ...

- **:help**
- vimtutor
- <http://inside.github.io/vim-presentation>
- <https://openvim.com>
- <https://vim.rtorr.com>