

GET USED TO COMMAND LINE INTERFACE

CREATED BY [@JUNHOCHO](#)

Start

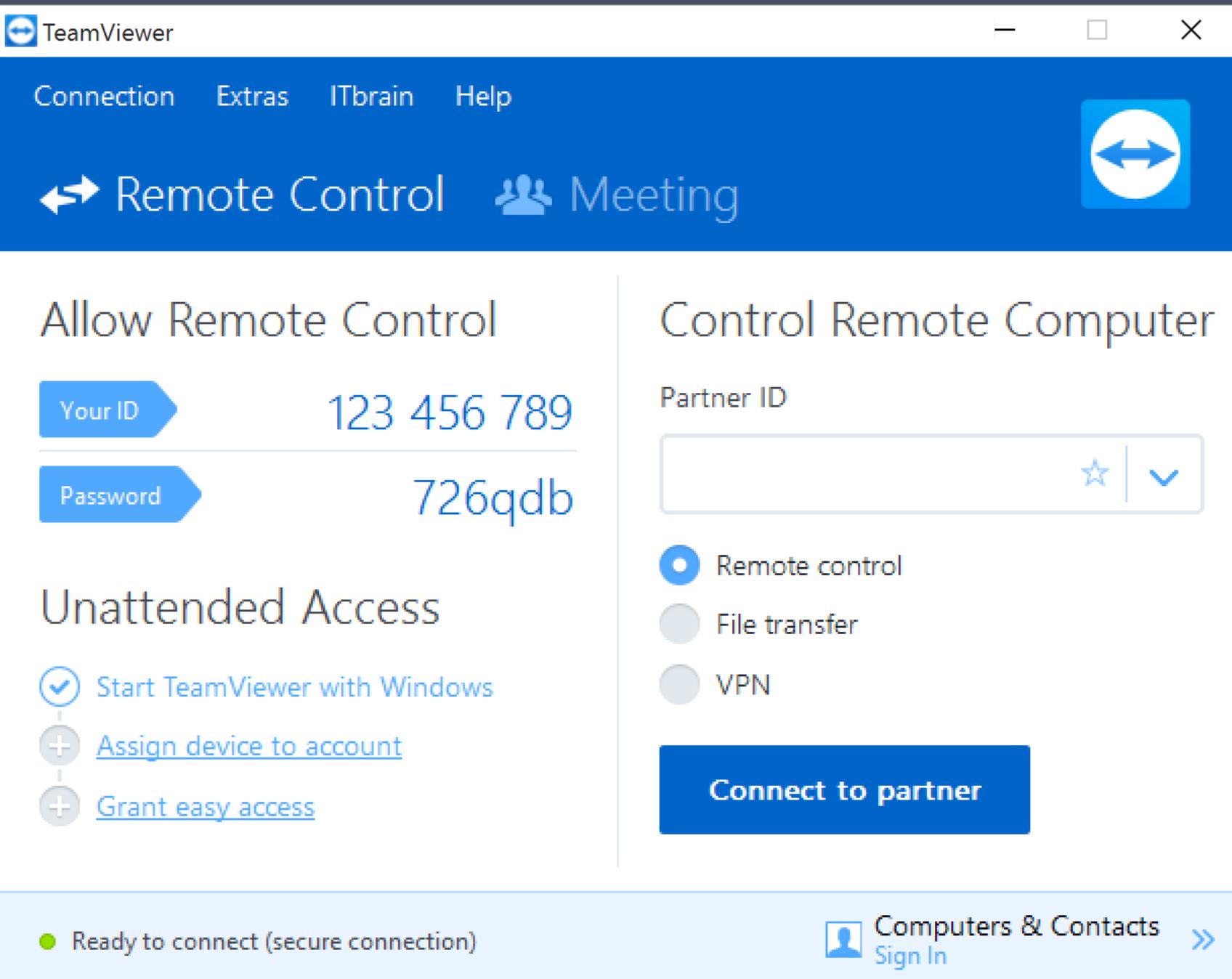


HEY. ISN'T GUI
IS ENOUGH?

WELL..



You know nothing, Jon Snow.



SATISFIED?

cli (TMUX + VIM)

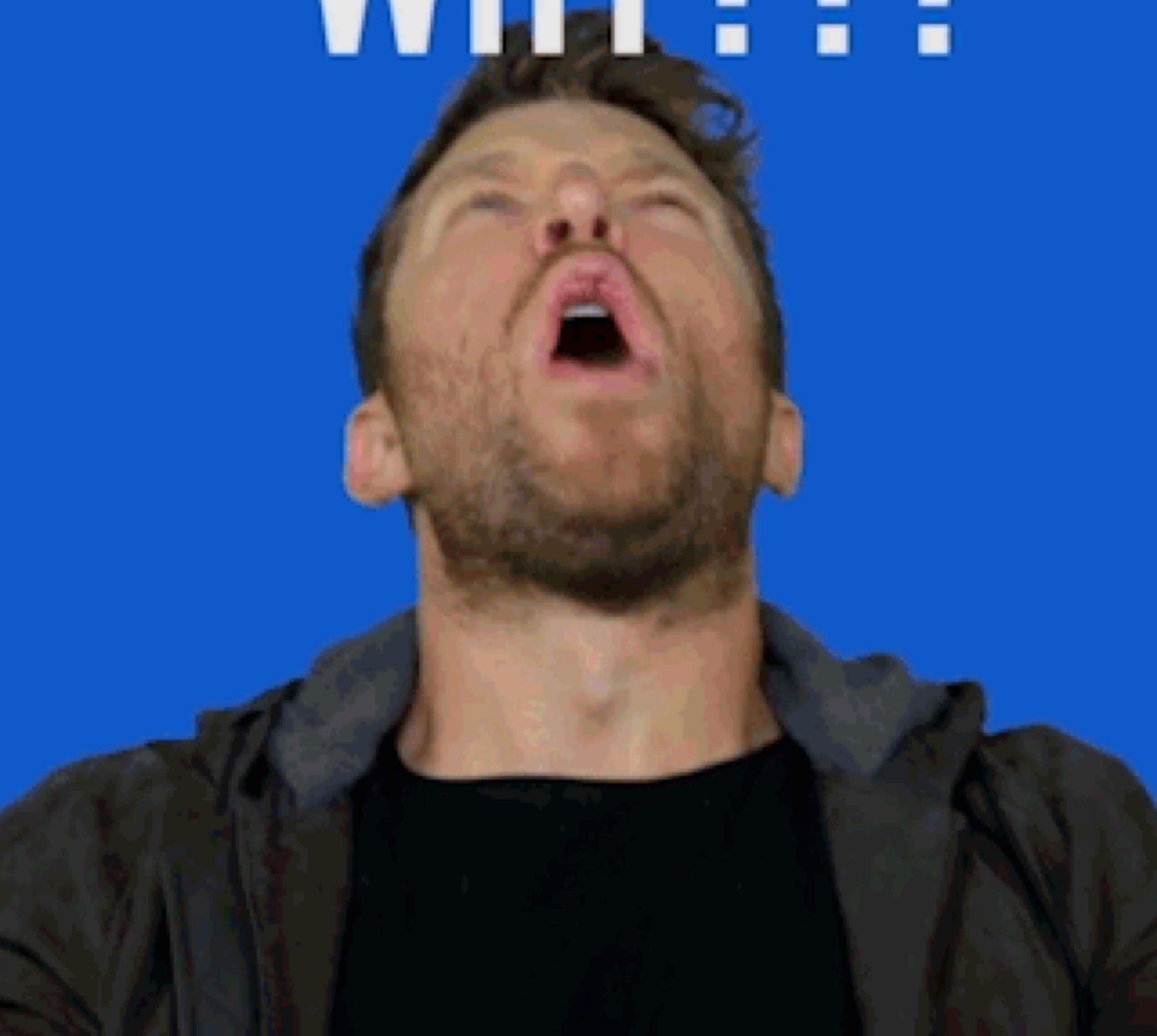
A screenshot of a tmux session with several windows and panes. The main title bar says "CLI (TMUX + VIM)".

- Panels:** The session has two panels. The left panel shows a file browser with a tree view of "Bilder", "screens", "doc", and "awesome-tmux". The right panel shows a terminal window with a file list and a command prompt.
- Windows:** There are four windows visible:
 - Window 1: Terminal, command: "cd projects > blog > public > Images > master > \$ grab".
 - Window 2: Terminal, command: "cd projects > blog > public > Images > master > \$ sleep 2".
 - Window 3: Terminal, command: "cd projects > blog > public > Images > master > \$ sleep 2".
 - Window 4: Terminal, command: "cd projects > blog > public > Images > master > \$ sleep 2".
- Bottom:** A tmux status bar at the bottom shows "manage" and "tobias" followed by a list of numbers from 1 to 9.

WHY CLI?

PORTABLE. SIMPLE. AUTOMATED. AND COOL

WHY???





YOU OWN SERVERS

YOU HAVE MAC ~~KAKAO TALK~~

YOU ARE LAZY

YOU ARE A BLOGGER

YOU ARE A COMPUTER GEEK

TRY CLI



TEACH ME.

THIS PRESENTATION ARE INSPIRED BY @NICKNISI
YOUTUBE LINK

COVERAGE

1. UBUNTU14.04 (THE OS)
2. VIM (THE EDITOR)
3. TMUX (THE WORKSPACE)

DOTFILES LINK: <https://github.com/junhocho/junhosetting>

LET'S START
WITH SSH

SSH : SECURE SHELL

SECURE CONNECTION TO SERVER

HOST(SERVER) IP : 147.46.89.175

USER ID : bctjv-[yourname]

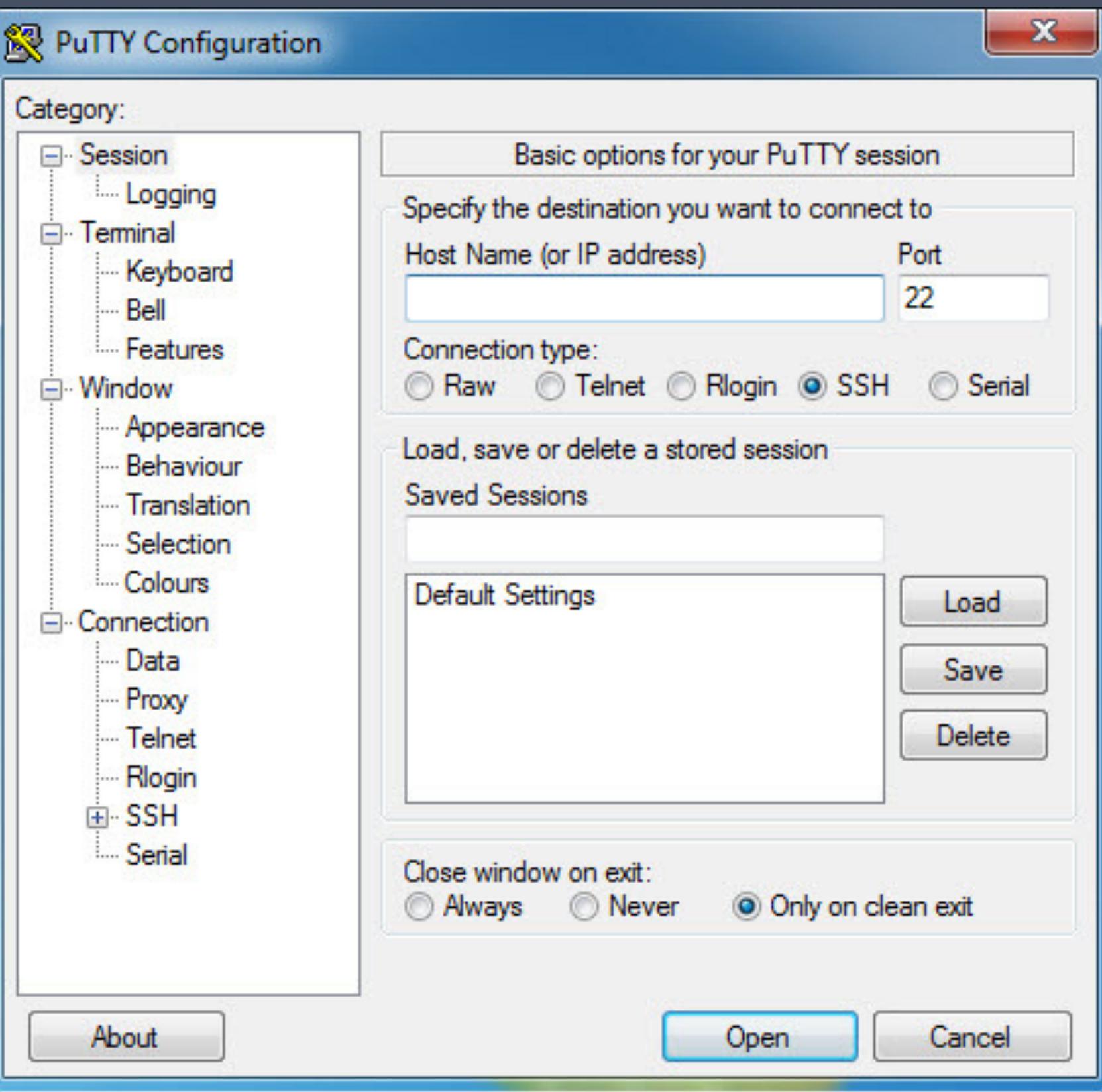
PASSWORD : bctjv



LOGIN !

LINUX/MACOS : ssh bctjv-[yourname]@147.46.89.175

WINDOWS : USE PUTTY



THIS ACTUALLY NEEDS

sudo apt-get install openssh-server

OPENSSH INSTALLED AT YOUR SERVER

AND ACTUALLY YOU DON'T NEED PASSWORD EITHER.
MAKE PASSWORD KEY AND KEEP IT !

REF

UBUNTU14.04 (LINUX)

- › ONE OF LINUX
- › COMMONLY USED IN DEEP LEARNING LIBRARIES
- › LINK ABOUT DIRECTORIES : TMP, ROOT, VAR, USR, HOME, BIN



ubuntu

SIMPLE COMMANDS

```
mv . cp . rm . mkdir . rmdir  
ls -lh. df -lh. du -sh . /*. whoami  
cp *.py /path/to/dest/  
rm -rf /다/지워/버리겠다/*.txt  
sudo apt-get install tmux
```

DO THIS NOW: git clone <https://github.com/junhocho/junhosetting>

```
[→ ~ cowsay HAHAHA!  
-----  
< HAHAHA! >  
-----  
 \ ^__^  
  \ oo)\_____  
   (__)\ )\/\)  
    ||----w |  
    ||     ||  
[→ ~ ]
```

TRY

cowsay

cmatrix

FUNNY COMMANDS

SOME SIGNALS

```
# ===== Ex1) =====
```

```
$ yes yell!
```

```
<C-c> # Control - c. Interrupt!
```

```
# ===== Ex2) =====
```

```
$ python -c '
```

```
c = 0
```

```
while True: print c; c= c+1'
```

```
<C-z> : stop for the moment and backgorund!
```

```
$ fg # back on foreground
```

```
# ===== Ex 3) =====
```

```
$ python
```

```
<C-d> # exit
```

CONCEPT

› DIRECTORY AND FILE / PERMISSION, OWNERSHIP

```
$ ls -al          # a: all, l : list

drwxrwxr-x  4 junho junho    4096 12월  4  2015 Workspace
-rw-r--r--  1 junho junho     3007  4월   3  2016 .zshrc
-rwxr-x---  1 junho junho     822   9월  17  2014 hello.py
```

CHANGE PERMISSION : \$ chmod 755 hello.py

CHANGE OWNERSHIP : \$ chown userid hello.py

sudo command: BORROW SUDO PRIVILEGE AND COMMAND

DO NOT ABUSE SUDO

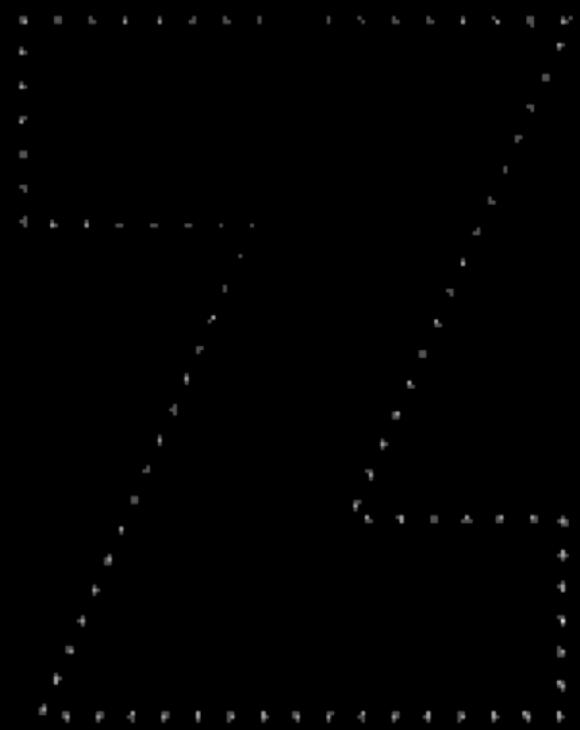


SHELL

- › DEFAULT SHELL IS bash shell
- › bash shell MADE IN 1989 WHICH IS PRETTY UNCOMFORTABLE.

USE zsh

- › MANAGE ZSH WITH [HTTP://OHMYZ.SH](http://ohmyz.sh)



```
Looking for an existing zsh config...
Found ~/.zshrc. ]Backing up to .zshrc.pre-oh-my-zsh
Using the Oh My Zsh template file and adding it to ~/.zshrc
Copying your current PATH and adding it to the end of ~/.zshrc for you.
Time to change your default shell to zsh!
Changing shell for eunsong
```

```
Password for eunsong:
```

1. SMART AUTOCOMPLETE

2. HIGHLY CUSTOMIZABLE

3. BUNDLED FUNCTIONS, HELPERS, PLUGIN, THEMES : EX) GIT

```
....is now installed.
Euns-MacBook-Pro% ^
```

INSTALL ZSH

```
sudo apt-get install git # git is very essential. Source code repositories  
sudo apt-get install zsh # Already installed in our case  
cd # move to home folder  
sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)" # install  
sudo chsh -s /usr/bin/zsh [your-id] # change your default shell into zsh
```

EASY?

LET'S DO SOME MORE TWEAK!

THEN. GIT CLONE Junhosetting FIRST

```
cd # move to home folder  
git clone https://github.com/junhocho/junhosetting #Star it  
cd junhosetting  
vi install.txt # peek vim
```

HEY, VIM USER

VIM : IMPROVED VI.

```
1 # Zsh setup
2 zsh autojump - https://github.com/wting/autojump.git
3 zsh syntax highlighting - https://github.com/zsh-users/zsh-syntax-highlighting.git
4
5
6 #alias-tips
7 $ cd ${ZSH_CUSTOM:-$ZSH/custom}/plugins          # 커스텀 플러그인 폴더로 이동
8 $ git clone https://github.com/djui/alias-tips.git      # 저장소를 로컬로 복사
9 $ $EDITOR ~/.zshrc                                # 에디터로 파일을 에디터로 불러들임
10 plugins=(git ... alias-tips) 추가
11 ...
```

LET'S DO SOME MORE TWEAK.

1. ZSH SYNTAX HIGHLIGHT

2. AUTOJUMP

VIM QUIT? ~~[ALT+F4]~~

```
:q      # vim quit  
:wq    # save and quit. identical to <shift-ZZ>
```

ZSH HIGHLIGHTING INSTALL

```
cd  
# get source of zsh-syntax-highlighting  
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git  
# put codes in .zshrc  
echo "source ${q-}PWD/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh" >> ${ZDOTDIR:-$HOME}/.zshrc  
# activate it now  
source ./zsh-syntax-highlighting/zsh-syntax-highlighting.zsh  
  
# this refreshes your .zshrc modofication to your current environment  
source ~/.zshrc # dotfiles locate in home folder
```

IT HIGHLIGHTS IF YOUR COMMAND IS VALID

DO AUTOJUMP BY YOURSELF

SOME MORE USEFUL COMMANDS

```
ls . | wc -l          # how many files in this directory.  
du -sh .*             # how big are folders and files in this directory  
df -lh                # list my Filesystem in human readable format  
ln -s [target-folder] [make-link] # make symbolic link  
grep -nr --color "keyword-to-find" * # locate "keyword-to-find"  
  
unzip some.zip        # unzip  
tar -xvf some.tar.gz # untar  
echo $CUDA_HOME        # print some environment variable  
wget https://goo.gl/ka7Yz5 # download file in web  
wget http://www.ekn.kr/data/photos/20150938/art_1442387799.jpg # Download jpg image  
python print-experiment-output.py 2>&1 | tee experiment.log # Store all print output in experiment.log  
which python           # useful to locate command  
  
history               # my command history  
pstree                # process tree  
htop                  # system monitoring  
nvidia-smi            # gpu monitoring  
who                   # which users are online?
```

ALIAS

> BE MORE LAZY.

YOU DON'T HAVE TIME TO TYPE: grep -nr --color "keyword-to-find" *

> PUT YOUR ALIAS AT THE END OF .zshrc

```
cd  
cat junhosetting/alias >> ~/.zshrc # Copy paste alias to end of .zshrc  
tail ~/.zshrc # print some lines of the end of .zshrc  
source ~/.zshrc # refresh this environment
```

```
grepn "key-word-to-find" * # Check the alias working
```

PORT

WELL KNOW PORTS : 22 : SSH / 80 : HTTP / 3389 : Window rdp ...

PORTFORWARD FROM CLIENT TO SERVER : ssh -L localhost:\$2:localhost:\$2 \$1

Ex) Do this at SERVER.

Jupyter notebook --no-browser --port 8888 # random ports 8888

Do this at CLIENT

ssh -L localhost:8888:localhost:8888 junho@147.46.89.175

Open web-browser in CLIENT and connect to `localhost:8888`

VNC (GUI) : USE GUI SAFELY WITH SSH

WEB-SERVER

```
# Do this at SERVER  
python -m SimpleHTTPServer 8888 # Open temporary web-server in 8888  
# Open web-browser in CLIENT and connect to 147.46.89.175:8888
```

SCP: MOVE FILE BETWEEN CLIENT AND SERVER

USE IT IN CLIENT (UNIX) TERMINAL.

BECAUSE CLIENT : 유동 IP / SERVER : 고정 IP

```
scp /client/A host:~/path/dest/B
```

Client의 파일 A를 host의 경로 B로 옮긴다.

```
scp host:~/path/dest/A /client/B
```

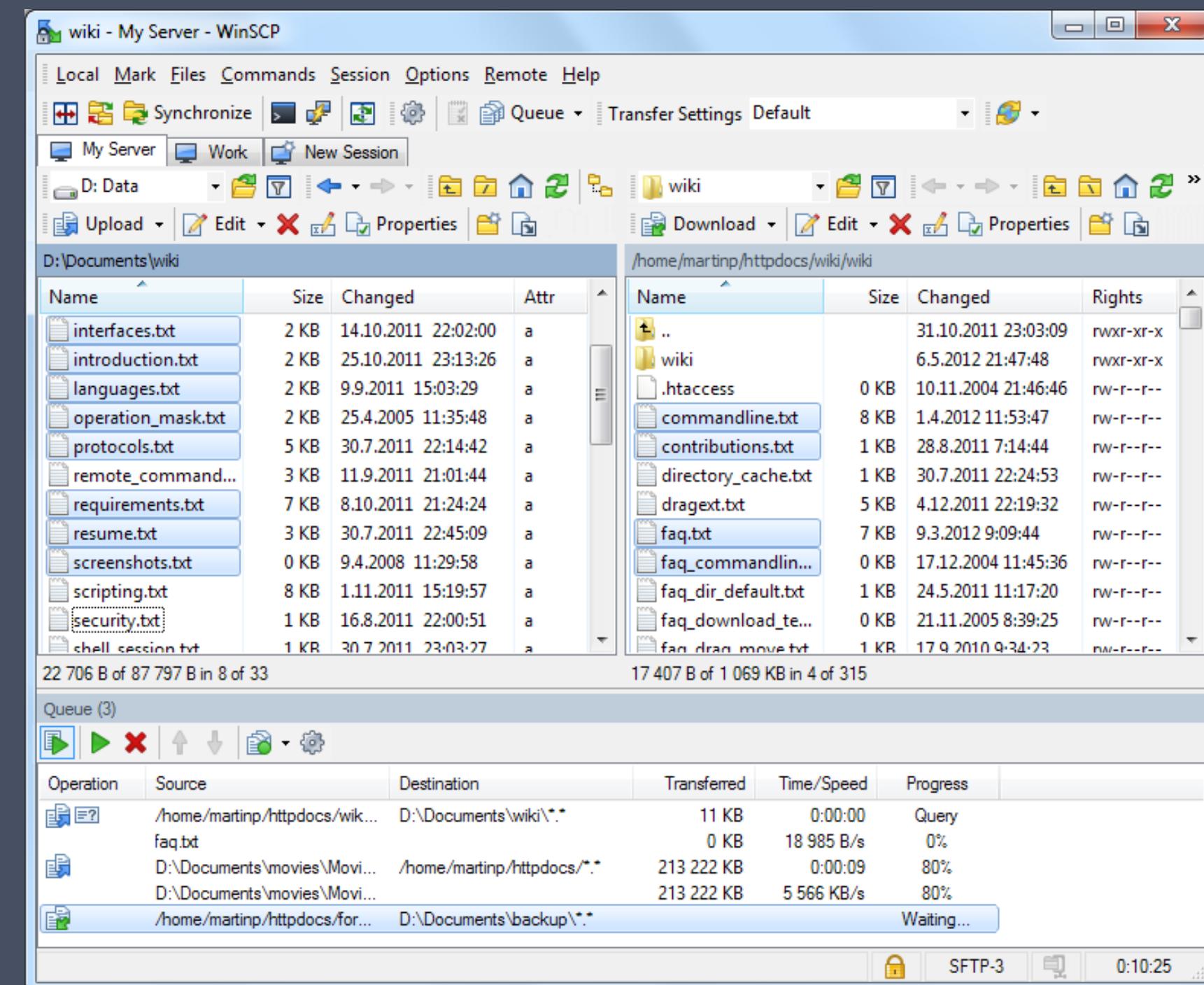
Host의 파일 A를 Client의 B로 옮긴다.

```
scp -r junho@147.46.89.175:~/Downloads/WordCount ~/Workspace/
```

-r 은 재귀적으로. 서버의 WordCount 디렉토리와 안의

모든 파일을 내 client의 Workspace의 복사

WINDOWS? WINSCP





imgcat

ONLY FOR MACOS TERM2

KILLING PROCESS

```
$ python -c '  
c = 0  
while True: print c; c= c+1'  
  
<C-z> # go to sleep and background.  
$ ps aux | grep python # print process that contains `python`
```

```
pil@pil-ubuntu:~/junhosetting/vnc$ ps aux | grep python  
pil      356 14.0  0.0 29676  7076 pts/45    T    01:44   0:00 python -c c = 0 while True: print c; c= c+1  
pil      398  0.0  0.0 17184  2536 pts/45    S+   01:44   0:00 grep python
```

kill -9 [process-id] : 356 IN THIS CASE

PROCESS KILLED!

ANY MORE QUESTIONS?



2. VIM



A screenshot of the Android Studio IDE. The top navigation bar shows tabs for 'Android', 'mobile', 'AppModule.java', 'wear', 'AndroidManifest.xml', 'AndroidManifest.xml', 'OpenWeatherApi.java', 'SweepWatchFaceService.java', and 'FlatFace'. The left sidebar shows the project structure with modules 'mobile' and 'wear', and Gradle Scripts. The main code editor displays Java code for a service, specifically the 'onDestroy' and 'onMessageReceived' methods. The bottom terminal window shows a single command: 'Alexs-MBP:FlatFace alexmullis\$'. The status bar at the bottom right indicates the Java version: 'JRE: 1.7.0_55-b13 x86_64' and 'JVM: Java HotSpot(TM) 64-Bit Server VM by Oracle Corporation'.

VISUA STUDIO . ECLIPSE. WEBSTORM. ANDROID STUDIO. PYCHARM. ZEROBANESTUDIO



I'D BE VIM

YOU DON'T HAVE CHOICE BUT TO LOVE VIM IN SSH.

VIM

- VIM : IMPROVED VI.
- PROBABLY ALREADY INSTALLED IN YOUR SYSTEM
- HIGHLY CUSTOMIZABLE
- SETTING FILE : .VIMRC

I Am Devloper
@iamdevloper

+ Follow

Over 1 year, the average Vim user saves 11 minutes in productivity.

RETWEETS 2,419 FAVORITES 1,619

7:55 AM - 15 Apr 2015

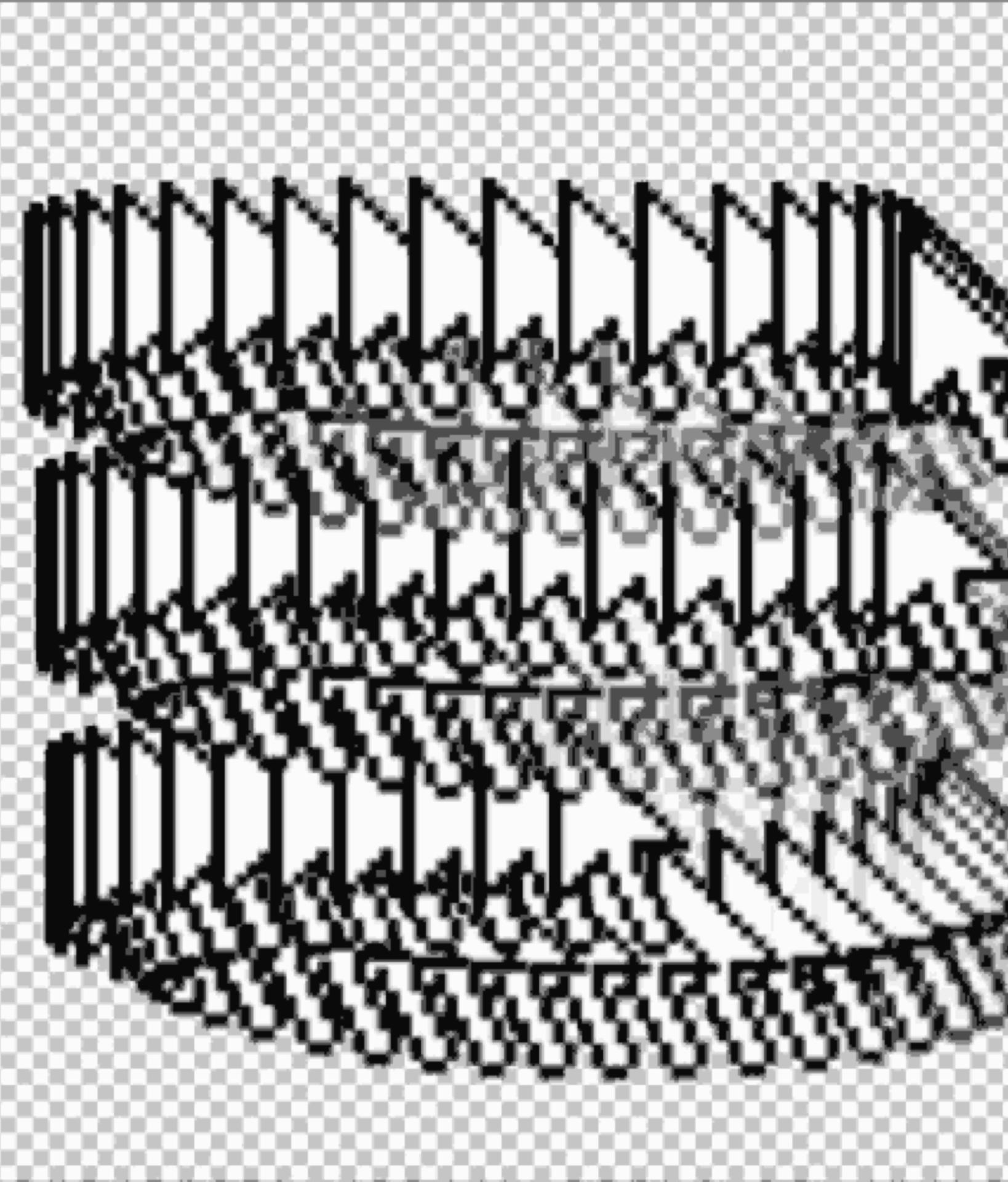
```
14      }-
13      top = rec;-
12      }-
11      ++top.count;-
10      roots[0].refs.push(top);-
9      }-
8      -
7      // remove classes with external references
6      while(roots.length){-
5          // hello world
4          // optimization: follow single linked list
3          refs = p.refs, refs.length = 1;-
2          top = [0];
1          if(!top || --top.count){-
54              // branch or end of chain => do not end tree
        roots-
1          top = 0;-
2          break;-
3      }-
```

MODAL EDITING

NORMAL ➔ ↵ master ➔ <re.js[+] javascript ➔ 4% ↵ 54: 89 ➔ * mixed-indent[229]

MODAL EDITING

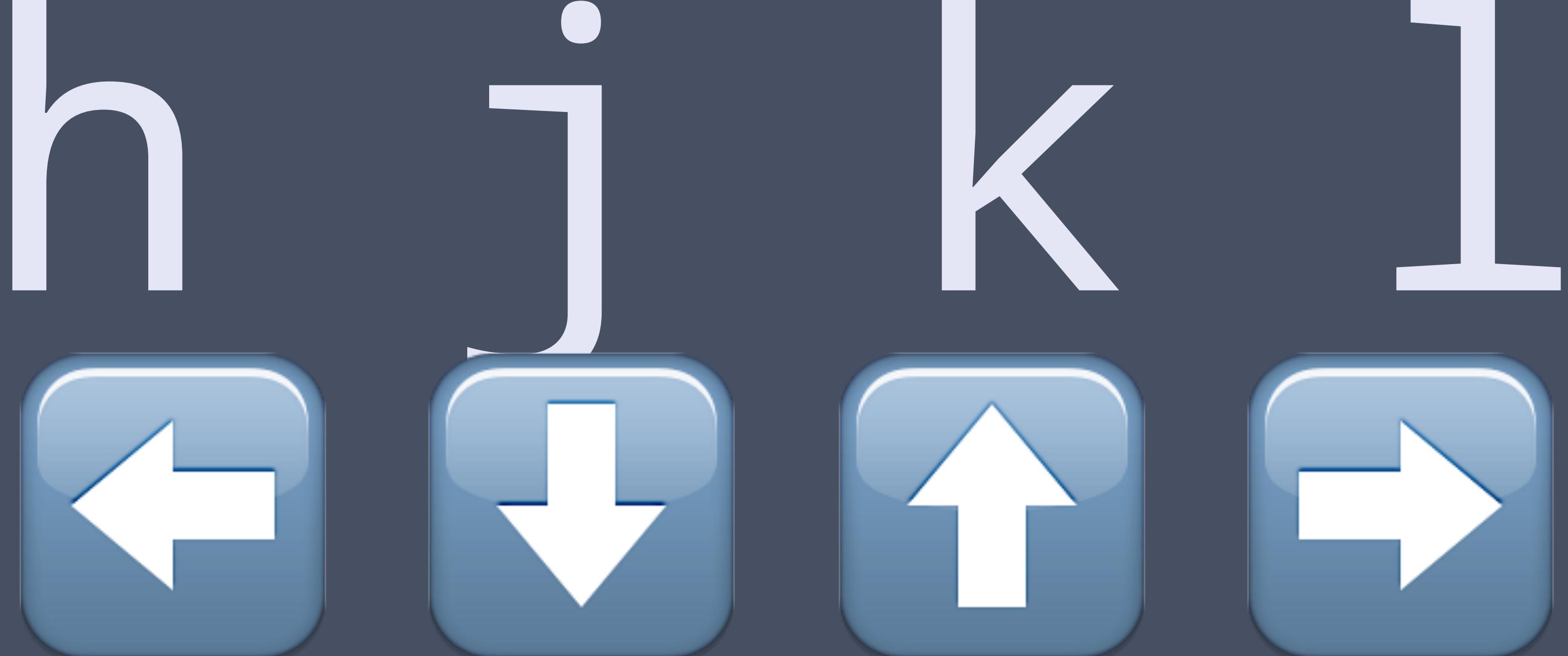
- > NORMAL - NAVIGATE THE STRUCTURE OF THE FILE <esc>
- > INSERT - EDITING THE FILE i, o, a, s, ...
- > VISUAL - HIGHLIGHT PORTIONS OF THE FILE TO MANIPULATE AT ONCE v, shift-v, ctrl-v
- > EX - COMMAND MODE :, /, ?



DON'T USE
MOUSE!



NOW ALL UPPER/LOWER ALPHABET DIFFERS



1 2 3

MOVE DOWN 12 TIEMS. NUMBERS AFFECT

- > :29: MOVE TO LINE 29
- > :set number: SHOW LINE NUMBERS
- > :set nonumber: no **USUALLY THE OPPOSITE OF COMMAND**

- > **$\wedge e$ - SCROLL THE WINDOW DOWN**
- > **$\wedge y$ - SCROLL THE WINDOW UP**
- > **$\wedge f$ - SCROLL DOWN ONE PAGE**
- > **$\wedge b$ - SCROLL UP ONE PAGE**
- > **H - MOVE CURSOR TO THE TOP OF THE WINDOW**
- > **M - MOVE CURSOR TO THE MIDDLE OF THE WINDOW**
- > **L - MOVE CURSOR TO THE BOTTOM OF THE WINDOW**

- > **zz : MOVE WINDOW SO THAT MY CURSOR IS AT CENTER**
- > **u : UNDO**
- > **^r : REDO**
- > **:w : SAVE**
- > **:q : QUIT**
- > **:wq : SAVE AND QUIT**

- > gg - GO TO TOP OF FILE
- > G - GO TO BOTTOM OF FILE

TEXT OBJECTS

- > w - WORDS
- > s - SENTENCES
- > p - PARAGRAPHS

MOTIONS

- > a- ALL
- > i - IN
- > t - 'TIL
- > f - FIND FORWARD
- > F - FIND BACKWARD

COMMANDS

- > d - DELETE (ALSO CUT)
- > c - CHANGE (DELETE. THEN PLACE IN INSERT MODE)
- > y - YANK (COPY)
- > v - VISUALLY SELECT

{command}{text object or motion}

DELETE IN WORD

CHANGE ALL WORD



YANK ALL TEXT INSIDE PARENTHESES

Va!!!

VISUALLY SELECT ALL INSIDE DOUBLEQUOTES
INCLUDING
DOUBLEQUOTES

^V 4j I # <esc>

VISUALLY SELECT COLUMN 4 DOWN LINES AND ENTER INSERT MODE TO WRITE #

COMMENTING MULTIPLE LINES WITH #

```
# Uncomment the following line to use case-sensitive completion.  
# CASE_SENSITIVE="true"  
  
# Uncomment the following line to use hyphen-insensitive completion. Case  
# sensitive completion must be off. _ and – will be interchangeable.  
# HYPHEN_INSENSITIVE="true"  
  
# Uncomment the following line to disable bi-weekly auto-update checks.  
# DISABLE_AUTO_UPDATE="true"  
  
# Uncomment the following line to change how often to auto-update (in days).  
# export UPDATE_ZSH_DAYS=13  
  
# Uncomment the following line to disable colors in ls.  
# DISABLE_LS_COLORS="true"  
  
# Uncomment the following line to disable auto-setting terminal title.  
# DISABLE_AUTO_TITLE="true"  
  
# Uncomment the following line to enable command auto-correction.  
# FIGNORE_CORRECTION="true"
```

REPETITION THE DOT COMMAND



A photograph of a person from the waist up, sitting on a light-colored wooden bench. The person is wearing a purple hoodie over a white t-shirt and camouflage-patterned pants. They are looking down at a black smartphone held in their hands. The background is slightly blurred, showing some greenery and a building.

REPEAT THE LAST COMMAND

THAT'S IT

A close-up photograph of a person's hand reaching through a jagged opening in a wall made of grey concrete blocks. The hand is gripping a piece of wood or metal, pulling it through the hole. The background is a dark, textured surface of the wall.

BUT IT'S POWERFUL

ADDITIONAL COMMANDS

- dd / yy - DELETE/YANK THE CURRENT LINE
- D / C - DELETE/CHANGE UNTIL END OF LINE
- ^ / \$ - MOVE TO THE BEGINNING/END OF LINE
- I / A - MOVE TO THE BEGINNING/END OF LINE AND INSERT
- o / O - INSERT NEW LINE ABOVE/BELOW CURRENT LINE AND INSERT

ADDITIONAL COMMANDS

- > J - DELETE LINE BREAK. PRETTY USEFUL
 - > p - PASTE
- > * ON WORD - FIND ALL THIS WORDS.
- > ^a ON NUMBER - INCREMENT

TRY AND MAKE ACTIONS REPEATABLE
PRACTICE IT

NOT EVERYTHING IS REPEATABLE
AT LEAST WITH THE DOT COMMAND



MACH 808

MACRO

A SEQUENCE OF COMMANDS RECORDED TO A REGISTER



RECORD A MACRO

- › q{register}
- › (DO THE THINGS)
- › q

PLAY A MACRO

- › @{register}

› mk : MARK POSITION AT k

`k : move cursor to position k

EX MODE

- > /someword : **FIND** someword
- > :spl / :vspl : **SPLIT (HORIZONTAL OR VERTICAL) VIM PANE**
 - > ^W + W/arrow : **MOVE PANE IN VIM**
 - > : **VERTICAL SPLIT PANE**
 - > :e. : **FILE TREE**
- > :%s/foobar/g : **REPLACE ALL** foobAR

- :syntax on :SYNTAX ON
- :set paste: PASTE MODE. USEFUL WHEN USING VIM
indent OPTION
- :set cursorline: SHOW CURSOR WITH UNDERLINE

LET'S PUT THOSE USEFUL OPTIONS
AS BASIC SETTING.

GET INTO JUNHOSETTING ONCE AGAIN

```
cp ~/junhosetting/vimrc ~/.vimrc
```

PLUGINS

PLUGINS : INSTALL WHAT YOU NEED

- > VUNDLE - PLUGIN MANAGER
- > NERDTREE - FILE DRAWER
- > CTRLP - FUZZY FILE FINDER
 - > FUGITIVE - GIT TOOL
- > SYNTASTIC - SYNTAX CHECKER / LINTER



BUT MY IDE DOES MORE

VIM IS EXTREMELY CUSTOMIZABLE
PLUGINS TAILOR VIM TO YOUR NEEDS

BUT WE CAN STILL DO MORE

NOT AN IDE
IT'S VIM

Shell

```
12   ▶   for (var i = 0; i < 5; ++i) { movable.update(); }-
11   ▶   expect(movable.get('x')).to.equal(val + 5);-
10   ▶   expect(movable.get('y')).to.equal(val);-
9    );-
8   ▶
7   ▶ it('should only allow speed to be settable to the MAX_SPEED', () => {-
6   ▶   movable.setDirection(Direction.East);-
5   ▶   movable.set('speed', Movable.MAX_SPEED + 1);-
4   ▶   movable.update();-
3   ▶   expect(movable.get('speed')).to.equal(Movable.MAX_SPEED);-
2   ▶   expect(movable.get('x')).to.equal(val + Movable.MAX_SPEED);-
1   ▶   expect(movable.get('y')).to.equal(val);-
85  ▶ });
1   ▶
2   ▶ it('should increment the rotation gradually', () => {-
3   ▶   var rotationSpeed = movable.rotationSpeed,-
4   ▶       newDirection = Direction.South,-
5   ▶       turns = newDirection / rotationSpeed;-
6   ▶   movable.set('direction', Direction.East);-
7   ▶   movable.set('speed', 1);-
8   ▶   movable.set('direction', newDirection);-
9   ▶   for (var i = 0; i < turns; i++) {-
10  ▶     movable.update();-
11  ▶   }-
12  ▶
13  ▶   expect(movable.get('direction')).to.equal(movable.rotation);-
14  ▶ });

```

NERD NORMAL ➔ ↵ typescript ➔ src/automaton/tests/unit/_Movable.ts typescript ← utf-8[unix] ← 85% ↵ 85: 6

"src/automaton/tests/unit/_Movable.ts" 100L, 3082C written

```
>> 3 non-emit-preventing type warnings
>> Type errors only.

TypeScript compilation complete: 1.33s for 15 typescript files

Done, without errors.
Completed in 1.851s at Mon Feb 23 2015 22:38:45 GMT-0600 (CST) - Waiting...

```

All files	90.32	55.56	90.22	90.23
~/code/automaton				

```
✖ typescript
❸ pasta 1 dotfiles 2 talk 3 dojo 4 automaton
❹ Bennie And The Jets - Elton John
22:38 23 Feb rugen
```

INSTALL VUNDLE WITH PLUGINS

```
cd  
git clone https://github.com/gmarik/Vundle.vim.git ~/.vim/bundle/Vundle.vim  
# clone in ~/.vim/bundle/ and setup your vimrc with plugins  
vi # then do :PluginInstall  
# Test :NERDTree with <F2> in vim  
# Test if autocomplete works
```

```
" Vundle  
set nocompatible          " be iMproved  
filetype on  
filetype off              " required!  
  
set rtp+=~/.vim/bundle/Vundle.vim  
call vundle#begin()  
  
" let Vundle manage Vundle  
" required!  
Plugin 'gmarik/vundle'  
Plugin 'tpope/vim-fugitive'  
Plugin 'Lokaltog/vim-easymotion'  
Plugin 'rstacruz/sparkup', {'rtp': 'vim/'}  
Plugin 'tpope/vim-rails.git'  
" vim-scripts repos  
Plugin 'L9'  
Plugin 'FuzzyFinder'  
" non-GitHub repos  
Plugin 'git://git.wicent.com/command-t.git'  
Plugin 'vim-ruby/vim-ruby'  
Plugin 'The-NERD-tree'  
  
Plugin 'Source-Explorer-srceexpl.vim'  
Plugin 'pyflakes'  
Plugin 'Python-Syntax'  
" Bundle 'Valloric/YouCompleteMe'  
Plugin 'AutoComplPop'  
Plugin 'nathanaelkane/vim-indent-guides'  
call vundle#end()          " required  
filetype plugin indent on  " required!  
"  
" Brief help  
" :PluginList      - list configured bundles  
" :PluginInstall(!) - install (update) bundles  
" :PluginSearch(!) foo - search (or refresh cache first) for foo  
" :PluginClean(!)   - confirm (or auto-approve) removal of unused bundles  
"
```

EXECUTABLES

WRITE ON TOP OF SCRIPT

`#!/bin/bash` IF SHELL SCRIPT.

`#!/bin/usr/python` IF PYTHON SCRIPT.

`chmod +x script`

THEN THE SCRIPT IS EXECUTABLE AS `$./script`

CTAGS SOURCE CODE INSPECTION

```
sudo apt-get install ctags
cd your-project
ctags -R . # will produce `tags` in the folder
vi code.cpp
< C - t > # to go inside declration
< C - ] > # to go back
```

3. TMUX

TERMINAL MULTIPLEXER

Shell

```

unit/-
  _Movable.js
  _Movable.js.map
  _Movable.ts
  all.js
  all.js.map
  all.ts
  Bot.js
  Bot.js.map
  [+]Bot.ts
  GameLoop.js
  GameLoop.js.map
  GameLoop.ts
  GameLoop.ts.map
  ObjectPool.js
  ObjectPool.js.map
  ObjectPool.ts
  Unit.js
  Unit.js.map
  Unit.ts
  Unit.ts.map
  [+]Unit.ts
  Bot.ts
  Bot.ts.map
  [+]Bot.ts

```

```

12   ▶   for (var i = 0; i < 5; ++i) { movable.update(); }-
11   ▶   expect(movable.get('x')).to.equal(val + 5);-
10   ▶   expect(movable.get('y')).to.equal(val);-
9    });-
8   ▶
7   ▶   it('should only allow speed to be settable to the MAX_SPEED', () => {-
6   ▶     movable.setDirection(Direction.East);-
5   ▶     movable.set('speed', Movable.MAX_SPEED + 1);-
4   ▶     movable.update();-
3   ▶     expect(movable.get('speed')).to.equal(Movable.MAX_SPEED);-
2   ▶     expect(movable.get('x')).to.equal(val + Movable.MAX_SPEED);-
1   ▶     expect(movable.get('y')).to.equal(val);-
85   ▶   );-
1   ▶
2   ▶   it('should increment the rotation gradually', () => {-
3   ▶     var rotationSpeed = movable.rotationSpeed;
4   ▶     newDirection = Direction.South;
5   ▶     turns = newDirection - direction / rotationSpeed;
6   ▶     movable.set('rotation', direction);
7   ▶     movable.set('direction', newDirection);
8   ▶     movable.set('turns', turns);
9   ▶     for (var i = 0; i < turns; ++i) {
10   ▶       movable.update();
11   ▶     }
12   ▶     expect(movable.get('direction')).to.equal(movable.rotation);
13   ▶   });
14   ▶ });

```

NERD NORMAL ➤ typescript ➤ src/automaton/tests/unit/_Movable.ts typescript ◀ utf-8[unix] ◀ 85% ↵ 85: 6

"src/automaton/tests/unit/_Movable.ts" 100L, 3082C written

```
>> 3 non-emit-preventing type warnings
>> Type errors only.
```

TypeScript compilation complete: 1.33s for 15 typescript files

Done, without errors.

Completed in 1.851s at Mon Feb 23 2015 22:38:45 GMT-0600 (CST) - Waiting...

1 pasta 2 dotfiles 3 talk 4 dojo 4 automaton

All files	90.32	55.56	90.22	90.23
-----	-----	-----	-----	-----

Done, without errors.

~/code/automaton

➤

x typescript

♪ Bennie And The Jets - Elton John

22:38 23 Feb rugen

WHY TMUX?

- > DON'T WANT TO STOP YOUR SSH SESSION.
 1. EXPERIMENT SESSION
 2. DOWNLOADING BIG FILE
 3. WEB-SERVER

- > EXTENDABLE WORKSPACE
- > AND MAINTAIN YOUR SESSION
- > CO-OPERATION

```
tmux new -s [session-name]      # create new session  
tmux ls                         # list existing session  
tmux attch -t [session-name]     # attach existing session
```

```
tmux new -s py-practice
```

<C-b> : TMUX BINDING-KEY.

- › WILL CHANGE IT TO <C-a> UNLESS YOU HAVE FLEXIBLE FINGER BONE

<C-b> c . bind c : NEW WINDOW

bind d : DETTACH FROM WORKING SESSION

LET'S CONFIGURE TMUX.

- › [REF1](#). [REF2](#)
- › [.TMUX.CONF](#)

```
cp ~/junhosetting/tmux.conf ~/.tmux.conf # dotfile로 이름바꿔서 복붙  
tmux source .tmux.conf # If you have working tmux session  
# Now your Tmux looks better and <C-a> is the binding-key
```

```
tmux new -s [session-name]
```

```
bind c:NEW WINDOW
```

```
bind n:NEXT WINDOW
```

```
bind p:PREVIOUS WINDOW
```

```
bind %:VERTICAL SPLIT PANE
```

```
bind "":HORIZONTAL SPLIT PANE
```

```
bind h(jkl, arrow):MOVE MY CURSOR PANE TO PANE
```

```
bind [ :COPY-MODE / q: COPY-MODE EXIT
```

```
exit / <C-d> :QUIT PANE
```

THERE'S A LOT THAT TMUX CAN DO



SYNCHRONIZE-PANES

bind :setw synchronize-panes

Shell

```
18 ~
17 # [fit] `tmux new-session -s {session-name}` ~
16 ~
15 ----
14 ~
13 # [fit] `tmux new-session -s pasta` ~
12 ~
11 ----
10 ~
9 # [fit] `~/.tmux.conf` ~ tmux configuration~
8 ~
7 # config tmux
6 ~
5 'bash'
4 # unbind default prefix and set it to Ctrl+a
3 Unbind Ctrl+
2 send-prefix
1 bind C-a send-prefix
342 ~
1 ~
2 ! [ -e /etc/vim/vimrc ] ~
3 ! [ -e /etc/vim/vimrc.local ] ~
4 ! [ -e /etc/vim/vimrc.bundles ] ~
5 ! [ -e /etc/vim/vimrc.bundles.local ] ~
6 ! [ -e /etc/vim/vimrc.bundles.vim ] ~
7 ! [ -e /etc/vim/vimrc.bundles.vim.local ] ~
8 # [fit] There's a lot that tmux can do~
9 ~
10 ----
11 ~
12 # [fit] Synchronize-panes~
13 ~
14 ! [ -e /etc/vim/vimrc ] ~
15 ~
16 ----
17 ~
18 # [fit] Create splits on the fly~
```

NORMAL → ↘ master vim-workshop.md
"vim-workshop.md" 360L, 5465C written

CREATE SPLITS ON THE FLY

bind % AND bind ''

TMUX + VIM = ❤

LINUX + GIT + TMUX + VIM = ❤

LEARN YOUR EDITOR WELL
(EVEN IF IT'S NOT VIM)

DON'T COPY SOMEONE ELSE'S CONFIGURATION
MAKE IT YOUR OWN

SETUP ALL CONFIGURATION

`./setup-all.sh # Do all the configuration we did today. Read it!`

ONESTEP!



BUT SHARE YOUR CONFIGURATION



DOTFILES

- › SHARE YOUR CONFIGURATION
- › STEAL IDEAS FROM OTHERS
- › JUNHOCHO/DOTFILES

OTHERS

- › NICKNISI/DOTFILES
- › BRYANFORBES/DOTFILES
- › JASONOX43/DOTFILES

KEEP PRACTICING



DOTFILES AND MATERIALS AVAILABLE AT [@JUNHOCHO](#)*

* INSPIRED FROM [@NICKNISI](#)