

Your Codebase:

How to get to where you want to be in it.

Max Wang | max.wang@college

vim

- “modal editing”
- Normal mode
 - Optimizes for movement and editing
 - Modular commands
- Insert mode
 - Time spent coding >> time spent typing

vim

- You don't have to use vim
 - Do use an editor with powerful features
 - Do good at using your tools
- Languages are a kind of devtool

vim basics

- **h/j/k/l** to move ←/↓/↑/→
- **i** for Insert mode; **Esc** for Normal mode
- **A** to append to a line
- **:w** to save, **:q** to quit
- **:!** for shell commands

vim basics

- **p** (paste), **r** (replace), **f** (find), **.** (repeat)
- operators: **d** (delete), **c** (change), **y** (yank), ...
- motions: **w**, **5j**, **t_**, **i{**, ...

vim basics

- `:help`

vim resources

- `$> vimtutor`
- <http://www.viemu.com/a-why-vi-vim.html>
- <http://zmievski.org/files/talks/codeworks-2009/vim-for-php-programmers.pdf>

Moving Within Files

Moving around

- **C-f** to page forward, **C-b** to page backward
- **gg** to go to top, **G** to go to bottom
- **/...** to search for a regex (weird syntax)
- **nG** or **:n** to go to line *n*
- **%** to find matching delimiter
- **gi** to go back to where you last inserted

Marks

- `m<letter>` to set a mark at cursor
- ``<letter>` to return to it
- `'<letter>` to return to the line
- Use viminfo to return to where you left off after quitting

Splits

- `:vsp` to create vertical split
- `:sp` to create horizontal split
- `C-w h/j/k/l` to move between them

Moving Between Files

Buffers and splits

- `:e` to open a new file
 - `:sp`, `:vsp` are similar
- Files open in buffers
- `:ls` to view buffers
- `:w n` to switch, `:bd n` to delete

gf

- Jump to the filename under the cursor
- :set path=... to handle #include's
- C-o to go back (“out”), C-i to go forward (“in”)

ctags

- Run ``ctags -R`` from your toplevel
 - Maybe add a make rule
- `C-]` over a symbol to jump to its definition
- `C-w]` to jump to it in a new hsplit
- `C-t` to backtrack along the tag stack

Plugins

Plugins

- Add awesome features to vim
- Use pathogen (<https://github.com/tpope/vim-pathogen>) to manage plugins
- Install plugins in `~/.vim/bundle`

NERDTree

- vim plugin
- `:NERDTreeToggle` to open interactive directory tree in a vsplit
- `o` or `<Enter>` to expand a directory or open a file
- <https://github.com/scrooloose/nerdtree>

Fugitive

- vim plugin
- Integrates lots of git commands
- <https://github.com/tpope/vim-fugitive>

.vimrc

- <https://gist.github.com/mxw/4724820>
- See <https://github.com/mxw/dotfiles/blob/master/vimrc> for my full config
- It's heavily documented!

Wrap-up

The bottom line

- **What you *use* to code** is important for the same reason **what you code *in*** is important
- Don't just skate by with nano and cd