A day in the life of a (ordinary) Vimmer

Presented by OKURA Masafumi
At VimConf 2018, 2018/11/24

VimConf🎊

All of us use (Neo)Vim everyday.

Question:
What’s a typical day of (ordinary) Vimmers?🤔

pp self

- Name: OKURA Masafumi
- Profession: Rubyist for 5 years
- Vim experience: 5 years
- Vim ability: Ordinary Vimmer <= Important!

About this presentation:

- What: I’ll show you how I use Vim in three cases.
- For whom: Beginner to intermediate, but experts will also take something away.
- How long: 20 minutes including demos.

Three use cases of (Neo)Vim

- Daily local development (in my case, Rails development)
- Code Reading
- Editing/viewing files in remote servers

Morning:
Daily local development

Why not modern editors?

- Atom is a great editor, but its Vim plugin doesn’t work as expected most of the time.
- VSCode is also a great editor, which I’d like to use if Vim doesn’t exist. However, default features are just too much.
- Emacs is just too difficult for me :(

Why not IDEs?

- I am a Ruby developer and there is only one relevant Ruby IDE: RubyMine.
- RubyMine is not open source or free. (However I'm interested in its unique features such as type annotations.)
- For other languages, it's a matter of choice. (However vim-go is so awesome that you might not need IDEs for go development.)

Daily startup routine

- Upgrading Vim and NeoVim
- Upgrading all Vim plugins with **:PlugUpdate** by **vim-plug**
- Seeing cow’s quote by **startify**

Plugins I use everyday

- **vim-rails**, must have for rails developer
- **deoplete** and **UltiSnips** for auto completion and high speed editing
- **ale** for on-the-fly lint

Plugins I use everyday (cont.)

- **vim-test** for agile testing
- **fugitive**, **gitgutter** and **GV** for Git operations
- **fzf** and **fzf.vim** for fuzzy search

Demo1

Afternoon:
Code reading

Vim as a code reader?

- Vim is a great tool to jump through files to files, code to code.

- With ctags, Vim gets the functionality to jump to method definitions.

- Vim has a rich set of search tools including internal grep, but you can use external tools like ag (the silver searcher).

Goodies for code reading

- **fzf.vim** provides us a handy command to search tags (**:Tags**) and do arbitrary search with ag (**:Ag**)

- ***** command (asterisk) searches string under a cursor, which is useful to find private methods.

- **Ctrl-]** command guides us to the tag under a cursor

Demo2

Remote servers

Notes about remote servers

- Vim is mostly installed in environments such as CentOS and Ubuntu.
- If we'd like to use NeoVim, first we need to build it in most cases which makes it hard to maintain.
- Vim version is often 7, or not the latest, but there is no problem.
- Sometimes it's impossible to install plugins.

Using Vim’s native features

- . command lets us repeat trivial editing such as inserting/changing/deleting objects.

- For more complex operations, there is a macro feature with **q** and **@** command.

- :argdo and similar commands are used to do bulk modification to files.

Useless tip:

```
vi='vim -N -u NONE -U NONE --noplugin --cmd "filetype indent on"'
```

Demo 3

Recap

- (Neo)Vim is a power tool for all kinds of software engineers.
- For local development, there are tons of useful plugins to help development.
- Vim’s native features are so powerful that without plugins we can do so much in unfamiliar places.

Learn once, use anywhere.

References

- My dotfiles is here: https://github.com/okuramasafumi/dotfiles

- Links to the plugins I mention in this slide are here: https://gist.github.com/okuramasafumi/5544889f4ddc711d1e0b108cf72e4e79