



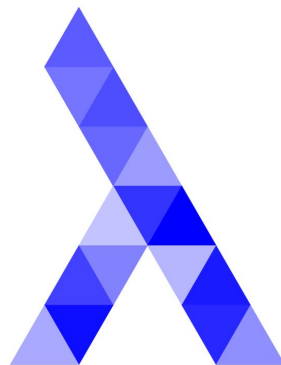
Modern techniques for implementing insert mode plugins

VimConf 2023 Tiny @kuuote



About me

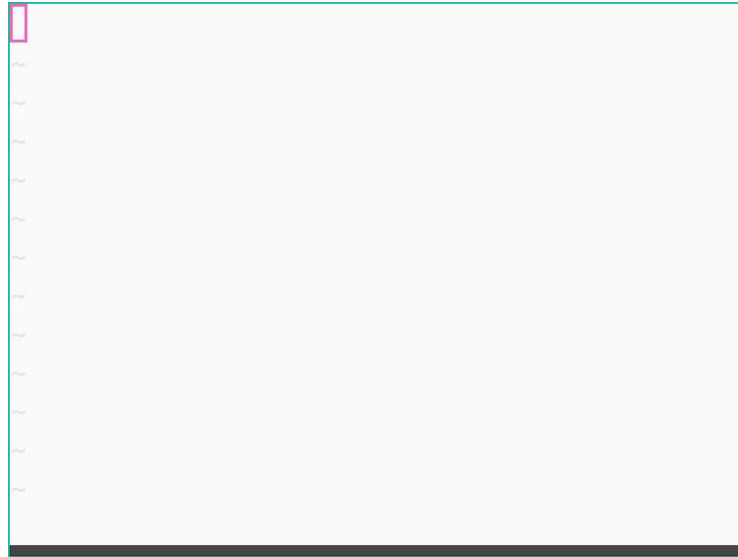
- kuuote (kuu + quote)
- icon is lambda
- Formerly Lisper(Schemer)
- Obsessed to Vim through making parantheses highlight plugin for Lisp
 - I want to rebuild it one of these days



:startinsert

- Typically text editor is always insert mode
- But Vim has normal mode
- In Vim, insert mode is a temporary mode which is somewhat special
 - Like inhaling & diving underwater (by ujihisa san)
- Unfit to complicated works

skkeleton



skkeleton

- One of SKK implementation
- SKK = Simple Kana to Kanji conversion program
 - Original from Emacs package called Daredevil SKK
- Some Vim implementation: `skk.vim` `skkim.vim` `eskk.vim`

skkeleton

- <https://github.com/vim-skk/skkeleton>
- Input method that not depends to system
 - Using Vim and Deno with denops.vim only

call function is hard

- Usual case for call function
- Normal mode
- Type `:` to Commandline mode
- Execute Ex command (call, execute etc...)

<Cmd> - execute command from everywhere

- Directly command execution at mapping evaluation
- `nnoremap <Space>e <Cmd>echo 42<CR>`
- `·:<C-u> (in normal) → <Cmd> (in any)`
- since Vim 8.2.1978
- since Neovim v0.3.0

<Cmd> - execute command from everywhere

- Before

- inoremap <expr> a 'hoge'

- After

- inoremap a <Cmd>call feedkeys('hoge', 'ni')<CR>

mapset()

- Can restore mapping
 - dumped by [maparg\(\)](#)
- since Vim 8.2.0807
- since Neovim v0.8.0

mapset()

- `let s:map = maparg('hoge', 'n', v:false, v:true)`
- `call mapset('n', v:false, s:map)`

mapset()

- `lua vim.keymap.set('n', 'hoge', function() end)`
- `echo maparg('hoge', 'n')`
→ `<Lua nn>`

popupwin / floatwin

- Display popup like completion menu
- popupwin since Vim v8.1.2384
- floatwin since Neovim v0.4.0
- Both API are similar and different

popupwin / floatwin

Before: echo

```
[No Name]                                0,0-1                                ALL  
a: 4 2 s: 四十二 d: 四二 f: # j: 42 k: XLII l: 肆拾貳
```

popupwin / floatwin

After: popup

```
4 2 |  
~ a: 4 2  
~ s: 四十二  
~ d: 四二  
~ f: #  
~ j: 42  
~ k: XLII  
~ l: 肆拾貳  
~
```

popupwin / floatwin

- See also: ddskk-posframe
- <https://emacs-jp.github.io/packages/ddskk-posframe>



Why using IME from Vim?

- OS and GUI are very complexity
 - Not many people can develop it
 - With Linux, especially
- Terminal and input methods don't work quite well together

Why using IME from Vim?

- Vim is portable platform
 - Can focus just on Vim users
 - Very easy to stabilize

How about trying it?

- In Vim, it's very comfortable
- Can control Vim mode naturally
- Customizable is happy

How about trying it?

- Out of Vim, too lazy
- It's a pain to copy and paste



Thanks for watching