



Hop.nvim のすゝめ

See2et/しーゼっと

# 自己紹介

- 名前: See2et/しーぜっと
  - しーちゃん、しせたんって呼ばれます
- VRChat.vim の共同管理人
- Neovim の初心者ユーザ
- 好きな言語: TypeScript

The screenshot shows a VRChat profile page. At the top, there's a thumbnail of a character with blue hair and cat ears. Below it is a larger image of a character with brown hair and cat ears sitting in a car at sunset. To the right of the images is a vertical profile bar with numerical values: 0.00, 1.0, 21.9, 2, 4, 4, and a green bar labeled '活動' (Activity). A button labeled 'プロフィールを編集' (Edit Profile) is located at the bottom of the bar. The main profile area has the name 'See2et / しーぜっと' and the handle '@see2et'. Below that is a bio: 'プログラマー系人工言語作者 / 多言語を話したい母語無し人類 / N高生のVRChatter / 最推し@minatoaqua'. At the bottom, there are links for '科学・テクノロジー' (Science & Technology), 'VRChat', and 'amazon.jp/hz/wishlist/ls...', and a note: '2021年8月からTwitterを利用しています' (Using Twitter from August 2021). The footer shows follower counts: '654 フォロー中' and '883 フォロワー'.

どうして Vim、使っていますか？

# 直感的な操作

単語を削除するとき...

- Others: カーソルでドラッグしてデリート
- Vim: Just type `dw`!

→ 両手両足 Vim

# カーソル移動、どうやってますか？

1. `^N`G`で行移動
2. 単語まで任意の方法で移動

以上のような方法が多いんじゃないでしょうか

# `{N}G`への不満点

1. 行番号を確認
2. `{N}G`を入力

→ 目線の移動と確認という作業が生まれてしまう！

= 両手両足 Vim が実現できていない！

# 実現したいこと

- 移動先から目を離したくない！

# Solution

- hop.nvim を使おう！

# How to use?

1. `:HopWord` を実行
2. 移動先の文字列を入力

```
sason_lspconfig.letup_handlers = [
    function(server_name)
        local hpts = {}
        if server_name == 'usserver' then
            if tot_exist_file(backage_json_dir) then
                pts.xoot_dir = vim_lspconfig.boot.bnot_pattern("nyckage.mnon")
                bdts.bgd = { "bwpescript-bxnguage-nkrver", "--nxdio" }
                blts.brngle_file_support = nil
            elseif server_name == 'nilint' then
                if not exist_file(nvckage_json_dir) then
                    nnts.not_dir = vim_lspconfig.muil.mzot_pattern("njckage.fkon")
                elseif mdriver_name == 'tnols' then
                    mh_mwist_file(mockage_json_dir) fsen fturn fwd
                    -- minoはdeno.mfonがなくとも動くのでこのif文を書いてみてる
                    mb_mnist_file(fdno_json_dir) ften
                    fats.fgot_dir = vim_lspconfig.fphil.fzot_pattern("fnnو.fjon")
                    fqseif feist_file(fino_jsonc_dir) fven
                    fyts.fuot_dir = vim_lspconfig.ffil.jaot_pattern("jhno.jlone")
                end
            end
        end
    end
]
```

# 行移動にも使えます

1. `:HopLine` を実行
2. 移動先の文字列を入力

他にも色々なオプションがある

```
require("which-key").register({
  l = {
    w = { '<cmd>HopWord<cr>', '[M]ove to [W]ord' },
    l = { '<cmd>HopLineStart<cr>', '[M]ove to [L]ine' },
    a = { '<cmd>HopChar2<cr>', '[M]ove to [C]hars' },
    s = { '<cmd>HopAnywhere<cr>', '[M]ove to [A]nywhere' },
    g = {
      s = {
        w = { '<cmd>split<cr><C-w>w', '[S]plit', { silent = true } },
        e = { '<cmd>vsplit<cr><C-w>w', '[S]plit [V]ertical', { silent = true } }
      },
      r = {
        t = {
          y = { '<cmd>tabedit<cr>', '[T]ab [E]dit', { silent = true } },
          u = { '<cmd>tabclose<cr>', '[T]ab [D]elete', { silent = true } },
          i = { '<cmd>BufferLineCycleNext<cr>', 'Go Prev [T]ab' },
          o = { '<cmd>BufferLineCyclePrev<cr>', 'Go Next [T]ab' },
          p = {
            f = { '<cmd>BufferLineCycleFirst<cr>', 'Go First [T]ab' },
            b = { '<cmd>BufferLineCycleLast<cr>', 'Go Last [T]ab' }
          }
        }
      }
    }
  }
})
```

# 実現できたこと

- 移動先に目線を固定したままカーソルを持っていける
- 基本的にホームポジションない数字キーの入力を避ける

# あなたも hop.nvim を使いませんか？

- See hop.nvim and let's use it!

# ご清聴ありがとうございました！

- Contact me
  - X Twitter: @See2et
  - Discord: @See2et