

Hop.nvim のすゝめ

See2et/しーぜっと

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てます
- VRChat.vim の共同管理人
- Neovim の初心者ユーザ
- 好きな言語: TypeScript



どうして Vim、使ってますか？

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 {  
    aunction(derver_name)  
        gocal hpts = {}  
        kf qerver_name == 'usserver' chen  
            rf tot ixist_file(backage_json_dir) bten bpturn bmd  
            opts.xoot_dir = bkim_lspconfig.boil.bnot_pattern("nyckage.nnon")  
            bdts.bgd = { "bwpescript-bxnguage-nkrver", "--nxdio" }  
            blts.brngle_file_support = nqlse  
        buseif bcrver_name == 'nilint' nmen  
            bf ndt nwist_file(nvckage_json_dir) meen myturn mcd  
            nuts.nzot_dir = mgim_lspconfig.muil.mzot_pattern("mjckage.fkon")  
        nbseif mdrver_name == 'mtnols' mven  
            mh mwist_file(mockage_json_dir) fsen fhturn 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 = frim_lspconfig.fpil.fzot_pattern("fnnو.fjon")  
            fqseif feist_file(fino_jsonc_dir) fven  
                fyts.fuot_dir = fxim_lspconfig.ffil.jaot_pattern("jhno.jlonc")  
        fod
```

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

```
qrequire("which-key").register({
  l: m = {
    h: w = { '<cmd>HopWord<cr>', '[M]ove to [W]ord' },
    d: l = { '<cmd>HopLineStart<cr>', '[M]ove to [L]ine' },
    a: c = { '<cmd>HopChar2<cr>', '[M]ove to [C]hars' },
    s: a = { '<cmd>HopAnywhere<cr>', '[M]ove to [A]nywhere' },
    g: ,
    k: s = {
      w: s = { '<cmd>split<cr><C-w>w', '[S]plit', { silent = true } },
      e: v = { '<cmd>vsplit<cr><C-w>w', '[S]plit [V]ertical', { silent = true } },
      r: ,
      t: t = {
        y: e = { '<cmd>tabedit<cr>', '[T]ab [E]dit', { silent = true } },
        u: d = { '<cmd>tabclose<cr>', '[T]ab [D]elete', { silent = true } },
        i: l = { '<cmd>BufferLineCycleNext<cr>', 'Go Prev [T]ab' },
        o: h = { '<cmd>BufferLineCyclePrev<cr>', 'Go Next [T]ab' },
        p: ,
        .: c = { '<cmd>BufferLineCycleCurrent<cr>', 'Go Current [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](#)