EEIC Vim Meetup

2022年2月21日

by matsui54

自己紹介

- 松井晴輝(GitHub ハンドルネーム: matsui54)
- Vim歴: 2年弱
- 使っているエディタ: Neovim
- Vimの設定ファイルの行数:
  約2400（コメント・空白行除く）
- Vim関係の活動:
  - 自作プラグイン開発
  - プラグインへのプルリクエストなど
  - 本体にも関わっていきたい...

Vimに抱くイメージ

- 古い
- 不便
- ダサい
- VSCodeでよくね

⇒ 今日はそのイメージを変えたい...!

Vimのイメージその1 ダサい

視覚に訴えるプラグイン

- デフォルトの色は確かに微妙
  ⇒ カラースキーム

カラースキーム

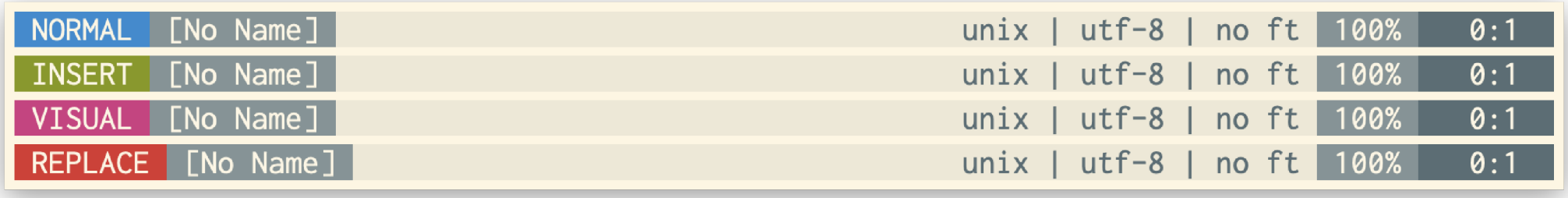
- iceberg: コントラスト低め
- gruvbox-material
- solarized: VSCodeにもあるやつ
- shirotelin: 白背景だけど見やすい
- その他たくさん...

colorswatというサイトで、カラースキームを見比べることができる。

ステータスライン

編集中のファイル、カレントディレクトリ、Gitブランチなどの情報表示

- lightline



https://raw.githubusercontent.com/wiki/itchyny/lightline.vim/image/solarized_light.png

ファイルのツリー表示

ディレクトリ構成を俯瞰

- ファイルの移動・削除などの基本操作
- 2画面ファイラー
- 一括リネーム

プラグイン

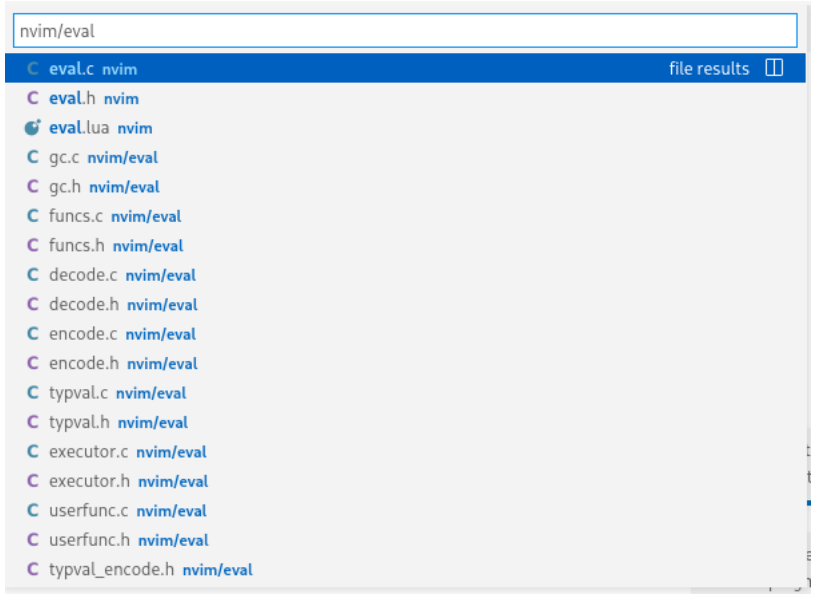
- defx.nvim
- fern.vim 初心者におすすめ。

プレビュー付きのファイル検索

カレントディレクトリ下のファイル検索、grep

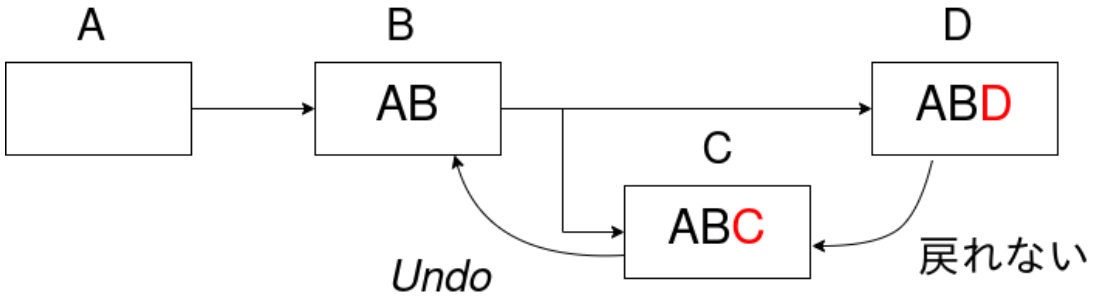
- ファジーファインダー
  - denite.nvim ddu.vimに開発は移行。設定は難しい。
  - fzf.vim 一番有名。
  - Telescope.nvim Neovim専用

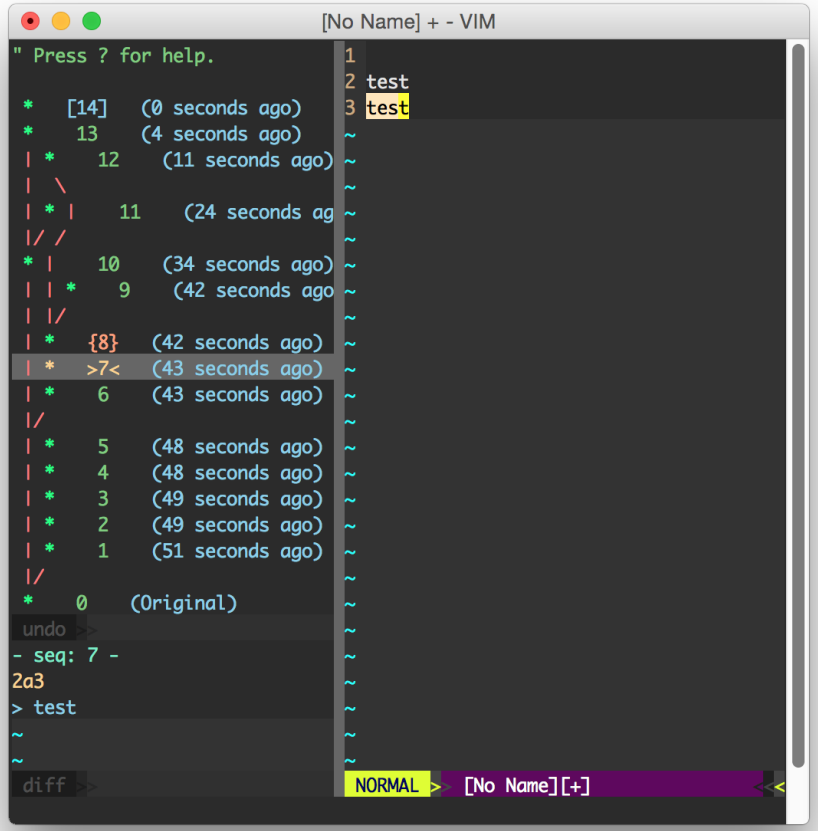
Vimにはファジーファインダーがたくさんあるので、興味のあるかたはこちらの記事をどうぞ。



その他視覚に訴えるプラグイン

- undotree 変更履歴の一覧表示。
- calendar.vim カレンダーを出せる。
- indentLine インデントを可視化する。





Vimのイメージその2 不便、IDE的な機能が使えない

Language Server Protocol (LSP)

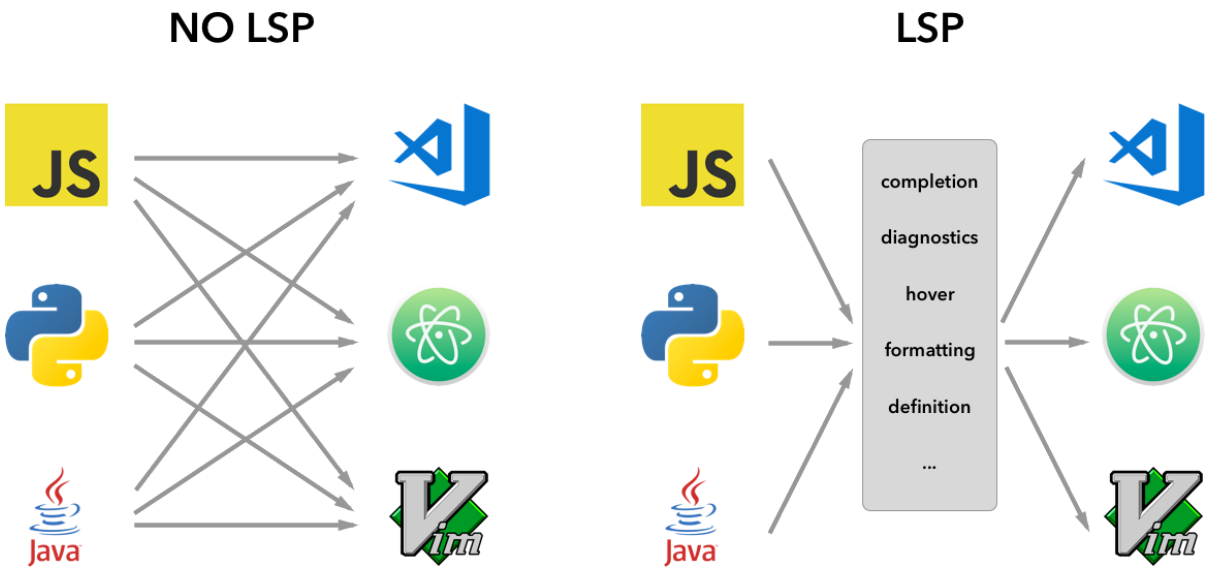
- 診断
- 定義ジャンプ
- 補完（引数・補完候補のドキュメントの表示）
- シンボル一覧
- 関数のヒントの表示（ホバー）
- コードの整形
- ...

といったIDEが標準的に備えている便利機能をまとめたプロトコルのこと。
このプロトコルはエディタ間で共通。
⇒ Vim/Emacs等でもLSP クライアントを入れればVSCodeのこれらの便利機能を使えるように!

Language Server Protocol (LSP) (おまけ)

昔は、便利な編集機能（定義ジャンプや関数や変数の補完）は言語ごとに実装され、そのツールごとにエディタが対応する必要があった。

LSPはこれらの便利機能をプロトコルとして定義し、エディタはLSPの仕様を実装すればどの言語のLanguage Serverの機能も使えるようになった。



https://code.visualstudio.com/assets/api/language-extensions/language-server-extension-guide/lsp-languages-editors.png

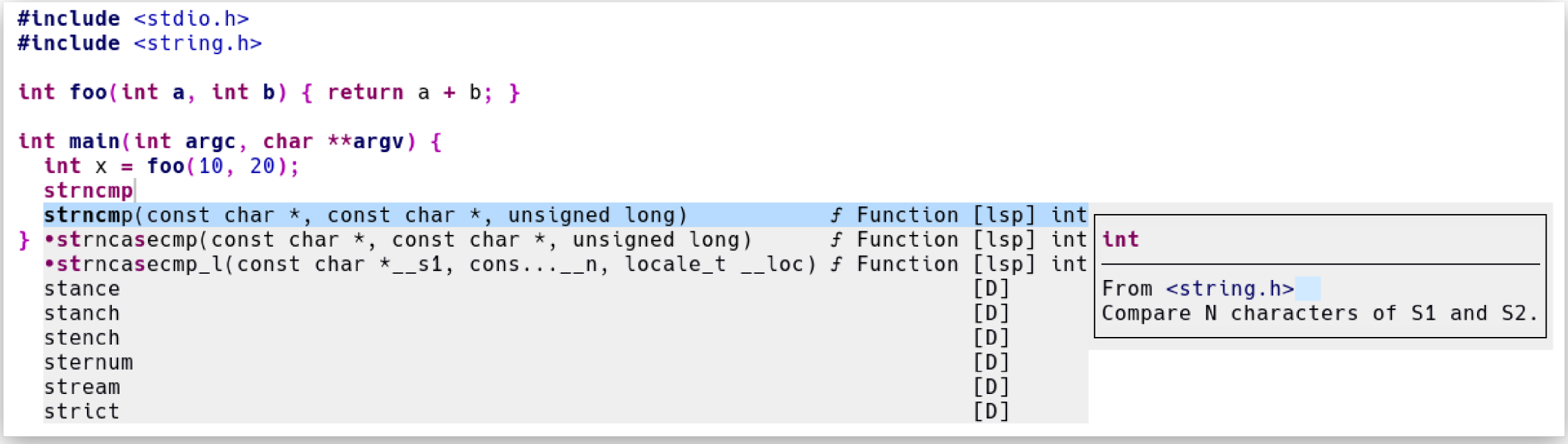
VimのLSPクライアント

- vim-lsp (Neovimではない) Vimでも動く。
- Neovim builtin LSP: Neovimの組み込みLSPクライアント。動作が速いがLuaで設定する必要あり。lspconfigが必要。
- coc.nvim "Make your Vim/Neovim as smart as VSCode."がモットー。
  初心者におすすめ。自動補完機能も持っている、オールインワンプラグイン。
  Nodejsを入れる必要あり。

自動補完

ファイル、辞書、周囲の単語などがサジェストされる。

- ddc.vim 最強のカスタマイズ性。Denoを入れる必要あり。
- coc.nvim 初心者におすすめ。
- nvim-cmp Neovimが使うならおすすめ。



スニペット

定型文を挿入してくれる機能。

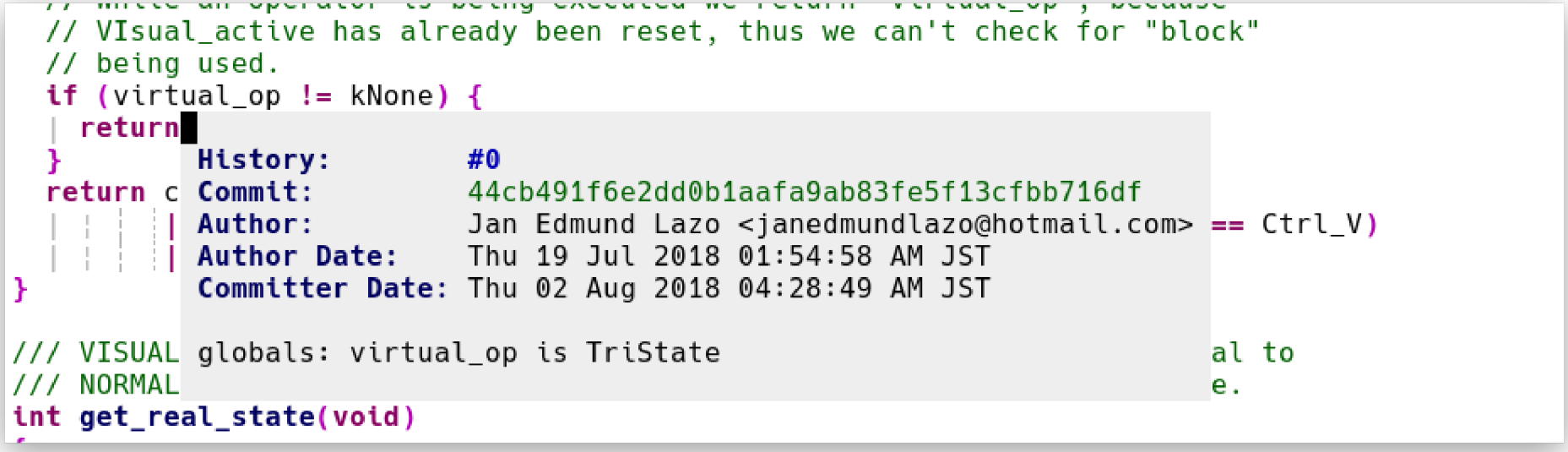
- vim-vsnip スニペット定義はVSCode方式。ミニマルで扱いやすい。
- ultisnips VSCodeよりもはるかに高機能。

```
// Cのfor文の例
for (${1:size_t} ${2:i} = ${3:0}; $2 < ${4:length}; $2++) {
  $0
}
```

```
% texでよく使うやつ
図\ref{$1}
```

Git操作

- gina.vim Git操作全般。ターミナルに戻らずに直感的に操作できる。
- git-messanger.vim 行ごとにコミット情報が出せる。



Vimの組み込み機能拡張プラグイン

- vim-sandwich ()や""で囲むoperatorを実現する。
- clever-f `f` コマンドを拡張。飛べる文字をハイライトしてくれる。
- vim-swap 関数の引数を一発で入れ替え。

その他便利なプラグイン

- vim-quickrun 一発でコンパイル&実行。
- caw.vim コメントアウトプラグイン。
- vim-tex texのプレビュー等。
- skkeleton 変態的な日本語入力機構であるSKKをVimで実現する。IMEを切り換える必要がない。

なぜVSCodeではなく、Vimを使うのか

- マウスを使いたくない (Vim拡張を使ってもマウス操作を強いられることがある)
- すべてを自分の思い通りにしたい。
  - VSCodeは拡張機能や設定で、ある程度ユーザーの思い通りにすることはできる。
  - しかし、エディタにもとから組込まれている機能については、不要な機能を削ったり、他の拡張に置きかえることはできない。
- プラグインの開発に参加しやすい。
  - VSCodeは巨大すぎて敷居が高い。

VimとNeovim

Vim

- 互換性重視
- どの環境でも入れやすい
- 一応新機能も追加されているが、なんか微妙...

Neovim

- モダンな機能がどんどん追加されている
  ex. LSP（VSCodeのような言語機能）, TreeSitter (よりよいシンタックスハイライト)
- Better defaults
- Lua（Vim scriptよりも数十倍速い）という言語で拡張可能
  （もちろんVim scriptも動く）

特に理由がなければ、Neovimを使うことをおすすめします。

Vimに興味を持った方へ

まずは実践Vimを読みましょう。
何度か出てきたcoc.nvimがVSCodeを普段使っている人には抵抗が少ないはず。

プラグインマネージャ

プラグインを入れてみたい人はこちら

- vim-plug 簡単でおすすめ
- dein.vim 起動時間をチューニングしたい人向け

おすすめ入門記事

「vim プラグイン」とかでGoogle検索上位に出てくるプラグインはすでに古くなっていることも多い...

- 上達したいVim初心者のための設定・プラグインの見つけ方、学び方
  基本的な心構えについて。記事内で言及されているvim-polyglotは現在はおすすめしません。
- 無人島に持っていく(Neo)vimプラグイン10選 (TS開発環境編)
- Neovim v0.5リリース記念 v0.5の新機能を紹介します【前編】
  Neovimについて知りたい方向け。

:wqall!