

# **The latest dark deno powered plugins**

Shougo

# Introduction

# Introduce myself



- <https://github.com/Shougo>
- Vim/Neovim patches/plugins author
- vim-jp member

# My old work for Vim/Neovim

- My Vim career **2008-2024**

(two thousand eight to two thousand twenty-four)

- Contribute for completions related features
  - `v:completed_item`, `getcompletion()`, etc...
- Developed plugins
  - `neocomplcache.vim`, `unite.vim`, `neobundle.vim`, etc...
  - `deoplete.nvim`, `denite.nvim`, `defx.nvim`, `dein.vim`, etc...

# About “real minimal” plugins

Gain the freedom to configure your plugins!



## Common scenario

- The author creates the simplest and minimal plugin.
- The number of users increases.



- But the author cannot accept the users' request.
- Or the plugin loses its simplicity due to added features.

What went wrong?



## **Regarding the current “minimal” plugins:**

- Minimal configuration?
- Easy to use?
- No, they are minimal only for the authors.
- They are not minimal for other users.



# **Why current “minimal” plugins are not minimal for other users:**

- Minimal means that there are no unnecessary features.
- People have different environments and minimal features.
- That is why you need to configure.

# Why don't you configure plugins?

- Prefer minimal configuration plugins?
- Tired of configuring plugins?
- No time to configure?

**Sigh...**



**Please remember this from today's presentation.**

# Your time exists to be used for the text editor!

カスタムレスポンス

Slackbot 15:16

「設定するのがめんどくさい」

<「すう.....設定を書くのがめんどくさい...?」

[vim-jp-radio #8 21:30](<https://audee.jp/voice/show/88533>)

**AuDee (オーディー) | 音声コンテンツプラットフォーム**

エンジニアの楽園 vim-jp ラジオ | 【Shougoさんが考えるプラグインのあるべき姿】 エンジニアの楽園 vim-jp ラジオ #8|AuDee (オーディー)

エンジニアコミュニティ「vim-jp」のラジオ版としてスタートした「エンジニアの楽園 vim-jp ラジオ」。

前回に続き、Vimプラグイン作者のShougoさんをお迎えます..... (379 kB)



<https://audee.jp/voice/show/88533>

# **Time spent using the text editor is free.**

The time spent on anything other than the text editor is wasted.

The text editor is not a means to an end in life but an end in itself.



## The “real minimal” plugins are ...

- Configure all yourself.
- Extensible.
- Do not work if you don't install/configure plugins.
- No side effects.
- Decoupling of implementation.

# The “minimal configuration” plugins example

<https://github.com/Saghen/blink.cmp>

```
1 lua <<EOF
2   local blink = require 'blink.cmp'
3   blink.setup()
4 EOF
```

# The “real minimal configuration” plugins example

<https://github.com/Shougo/ddc.vim>

```
1 call ddc#custom#patch_global(  
2 \ #{ ui: 'pum', sources: ['around'] })  
3 call ddc#custom#patch_global('sourceOptions', #{  
4 \   _: #{  
5 \     matchers: ['matcher_head'], sorters: ['sorter_rank']  
6 \   },  
7 \ })  
8 call ddc#enable()
```



# Commonly used plugins are too user-friendly

- Users are wise and omniscient.
- Users understand their own configurations.
- Copy-pasting settings is unacceptable.



## Q: So you need to study plugins before use?

- A: **Of course, yes!**
- Didn't you study Vim when you started using it?
- It feels uncomfortable when plugins work without any configuration.



# **The world is full of configurations:**

- Creating your own plugins and text editors.
- Using plugin distributions.
- Using Nix.
- Installing plugins.

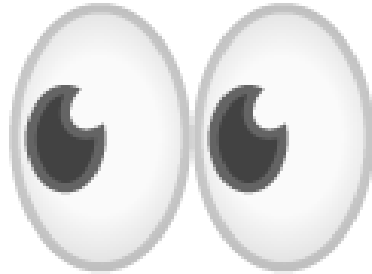
Isn't saying "I don't want to configure" the same as saying "I don't want to breathe"?

# My plugin development policy

- I want to fulfill **the wishes of all plugin users**.
- I don't consider users who **don't want to configure the plugin**.
- I need minimal reproducible bug report to fix bugs.
- No default mappings.
- No commands.

# The latest dark deno powered plugins

Be Awaken to the text editor!



# What means the “dark”?

- Please see `:help design-not`.
- I have defined it is the **light(right)** way.

# The “light” way



- Vim is not OS/shell/terminal.
- Vim is not same with Emacs.
- Vim is not everything.

# The “dark” way



- Vim is OS/shell/terminal.
- Vim is same with Emacs.
- Vim is everything.



# The latest dark deno powered plugins

- `ddc.vim`
  - <https://github.com/Shougo/ddc.vim>
- `ddu.vim`
  - <https://github.com/Shougo/ddu.vim>
- `dpp.vim`
  - <https://github.com/Shougo/dpp.vim>

# Deno powered technology



- `denops.vim`
  - <https://github.com/vim-denops/denops.vim>
- Please see lambdalisue's Vimconf Tiny presentation
  - <https://slides.vimconf.org/2023/02-alisue-denops-v2.pdf>

# Configurable using TypeScript

```
import type { Denops } from "jsr:@denops/std@7.2.0";
import * as fn from "jsr:@denops/std@7.2.0/function";

import { expandGlob } from "jsr:@std/fs@1.0.0/expand-glob";

export class Config extends BaseConfig {
  override async config(args: {
    denops: Denops;
    contextBuilder: ContextBuilder;
    basePath: string;
  }): Promise<ConfigReturn> {
    const hasNvim = args.denops.meta.host === "nvim";
    const hasWindows = await fn.has(args.denops, "win32");

    const inlineVimrcs = [
      "$BASE_DIR/options.rc.vim",
      "$BASE_DIR/mappings.rc.vim",
      "$BASE_DIR/filetype.rc.vim",
    ];

    if (hasNvim) {
      inlineVimrcs.push("$BASE_DIR/neovim.rc.vim");
    } else if (hasNvim) {
      if (await fn.has(args.denops, "gui_running")) {
        inlineVimrcs.push("$BASE_DIR/gui.rc.vim");
      }
    }

    if (hasWindows) {
      inlineVimrcs.push("$BASE_DIR/windows.rc.vim");
    } else {
      inlineVimrcs.push("$BASE_DIR/unix.rc.vim");
    }

    args.contextBuilder.setGlobal({
      inlineVimrcs,
      extParams: {
        installer: {
          checkDiff: true,
        },
      },
    });
  }
}
```

- You can configure plugins by TypeScript.
- You can use the type information.

# TypeScript configuration example

```
1 export class Config extends BaseConfig {
2     override async config(args: ConfigArguments):
3     Promise<void> {
4         args.contextBuilder.patchGlobal({
5             ui: "pum", sources: ["around", "file"],
6             sourceOptions: {
7                 _: { ignoreCase: true, matchers:
8                 ["matcher_head"], sorters: ["sorter_rank"]},
9                 ...
10            }
11        }
12    }
```

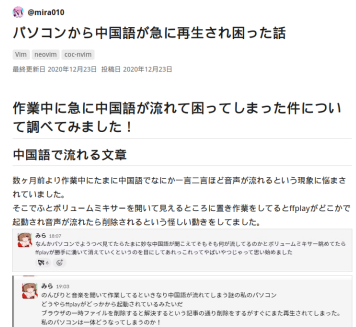
# Extensible using TypeScript



- You can create the extensions by TypeScript.
- Please see:
  - <https://github.com/topics/ddc-source>
  - <https://github.com/topics/ddu-source>
  - <https://github.com/topics/dpp-ext>

# No side effects

- They do not work if you don't configure plugins.
- If you install side effect plugins, you may get unknown sound from Vim...
  - ▶ <https://qiita.com/mira010/items/ff73bdb5d922bc89c021>



# No default mappings

- You can define your original mappings.
- You don't need to know what is the default mappings.
- You don't need to disable the default mappings.
- You don't worry about plugin conflicts.

# No commands

- You can define your original commands.
- Functions are better for me.
- The commands wrapper is available for ddu.vim.
  - <https://github.com/Shougo/ddu-commands.vim>



# **Decoupling of implementation**

- You can only use the enabled features that you need.
- You don't need to worry about what features are installed.
  - Because you need to install only what you need!

# The plugins are complex?

- **No.**
- Because the configuration is really simple.
- You just need to **configure everything**.



## **But...**

- The plugins can fulfill **any user's wishes**.
- If you trying to make the plugin do complex things will make the configuration complicated.

# Conclusion

- The dark deno powered plugins are **“real minimal”** plugins.
- Your time exists to be used for the text editor.
- You need to study Vim and plugins.

# Additional Information

- `ddt.vim`
  - Dark Deno Terminal interface for Vim/Neovim
  - The next of `vimshell.vim/deol.nvim`.
  - <https://github.com/Shougo/ddt.vim> (Under construction)

# **You can support the real minimal plugins development**

- <https://github.com/sponsors/Shougo/>
- The pull request and issues are wellcome.

**End**

Thank you for your attention!