

LaTeX 文書作成における代替エディタの検討

[ru_museum](#): GitHub)

2025 年 10 月 29 日

目次

1	概要	1
2	作業環境	1
3	Editors	1
3.1	KDevelop	1
3.2	Kate	2
3.3	Pluma	3
3.4	Zed	3
3.5	VSCodium	3
3.6	VSCode	4
4	TIPS	4
4.1	PDFViewer	4
4.2	連続するハイフン (余録)	5

1 概要

- これは **Gedit** の不具合^{*1}による、他エディタへの一時的移行に関する手引きです。
- **Gedit** 自体がプラグインに関し Python の非推奨とかが検討され、**C** 言語での再開発が試みられており、各種不具合が生じ機能の低下が心配されています^{*2}。
- **Gedit** の fork である **Pluma** も **Snippets** が動作しません。
- 軽量で使い勝手の良い **Gedit** には一日も早く元に戻って欲しい所ですが、**LaTeX** を扱う上で何れが次善であるかを検証しています。

2 作業環境

- GNU/Linux Debian 6.16
- Lua \LaTeX / LuaHB \TeX 1.22.0 (TeX Live 2025/Debian)
- KDevelop(v6.1)、Pluma(v1.26)、Kate(v25)、VSCodium(v1.105)、Zed(v0.206.7)

3 Editors

3.1 KDevelop

- **KDevelop** は KDE のテキストエディタ「Kate」と内部的に共通しています。

3.1.1 External Script

- [設定] \Rightarrow Kate を設定 \Rightarrow **External Tools** で追加・削除・編集を行います。

設定例 (LuaLatex 使用)

設定項目	入力値
Name	Lualatex Build
Command	lualatex --shell-escape %f && atril "%b.pdf"
Input	Nothing
Output	ignore
Errors	ignore
Save mode	Save Active Document
Show output	✓
Output Filter	No Filter
Shortcut	メイン：Alt+F5 代替：なし

^{*1} 以下のエラーにより起動自体が出来ないという不具合が生じています。様々試みていますが未だ解決に至っていません。

gedit:symbol lookup error:libgedit-48.1.so: undefined symbol:gtk_source_encoding_get_utf8

^{*2} gedit は Gtk4 への移行に伴い、最近の v48 から組込 plugin の **externaltools** と **snippets** を除外することとしています。

基本的に v48 から全ての Python 製のプラグインを除外しています。

詳しくはプラグイン自体の問題ではなく、PyGObject が最新バージョンでサポートされなくなった事が原因らしいです。

2025-04-30 :: No more Pythons in gedit.

Are Third party plugins still going to be a thing ?

- コンパイル後に PDF を開きます。PDFViewer には **Atril** をお奨めします ^{*3}。
- 「Shortcut：メイン」は自由に設定出来ます（組合せが必要）。
- 上記例 **Save mode: Save Active Document** の場合、コンパイル前の「保存」イベントは必要なく、「Alt+F5」(Shortcut) でのコンパイルと同時にファイルは保存されます。
- %f、%b 等の **Command** に使用するの設定値は以下を参照して下さい。

記述	内 容
%u	URL of the active document.
%f	local filepath to the active document.
%n	name of the active document, including its extension.
%b	name of the active document without its extension.
%d	path to the directory of the active document.
%p	URL to the project of the active document.
%s	shell escaped contents of the selection in the active document.
%i	PID of the currently running KDevelop process.

3.1.2 Snippets

1. 設定の Plugins から Snippets を有効にします。
2. Snippets に追加・登録します。

3.2 Kate

- **Kate** は、**KDevelop** の元となっていますが使い勝手が若干の異なりますので注意して下さい。

3.2.1 External Tools:

- [設定] ⇒ Kate を設定 ⇒ **External Tools** で追加・削除・編集を行います。
- **Executable** は、ここでは直接 **lualatex** ではなく **sh** を介して扱います。

Command は一行の Shell Script として、SH のコマンドオプション '**-c**' で行います。

記入例：-c 'lualatex --shell-escape "%{Document:FileName}" && atril "%{Document:FileName}.pdf"'

設定例 (LuaLatex 使用)

設定項目	入力値
Name	Lualatex Build
Executable	sh
Command	-c 'lualatex --shell-escape "%{Document:FileName}" && atril "%{Document:FileName}.pdf"'
Input	Nothing
Output	ignore
Errors	ignore
Save mode	Save Active Document
Show output	√
Output Filter	No Filter
(Shortcut)	「保存」と同時に発火されますので不要です。

^{*3} evince にはコンパイル時に CPU を過剰に占有しシステムを停止するという重大なバグがあります。

3.2.2 Snippets :

1. [設定] ⇨ Plugins ⇨ Snippets
2. 右側サイドバーのアイコンで表示・非表示を行います。
3. 自由に追加・削除・編集が可能です。

3.3 Pluma

- **Pluma** は **Gedit** から分岐したもので使い勝手とインターフェイスは大体同じです。
- プラグインの Externaltools と Snippets は一応ありますが、Snippets は動作しませんでした^{*4}。

3.4 Zed

- X11 環境では日本語入力不可（未対応：2025/10）。
- Wayland 環境では可能？（未検証）。

3.5 VSCodium

- MS VSCode から Fork し情報収集・追跡等を行う **telemetry/tracking** を **disabled** としたセキュリティ問題に対処したものとなっています。
- ROOT ユーザーでの使用：# codium --user-data-dir="/root/.config/Code" --no-sandbox
- Download: [VSCodium](https://github.com/VSCodium/vscodium/releases)(GitHub) //github.com/VSCodium/vscodium/releases

3.5.1 Task の作成

tasks.json の 新規作成と編集

Manage（左下ギアマーク）⇨ Tasks

Terminal ⇨ Configure Tasks

.config/Code/User/tasks.json

* 各 Variable の値を知るには**"command": "echo"** としてテストすることが出来ます。

【設定上の注意点】

- (a) 生成するファイルのフォルダ指定が必要：--output-directory='\${fileDirname}'。
- (b) 開く PDF にも PATH が必要："'\${fileDirname}/\${fileBasenameNoExtension}.pdf'"。

```
"tasks": [
  {
    "label": "LuaLatex Build",
    "type": "shell",
    "command": "luaLatex", // テスト用には "echo" とします
    "args": [
      "--shell-escape", "--output-directory='${fileDirname}'", "${file}",
      "&&", "atril", "'${fileDirname}/${fileBasenameNoExtension}.pdf'"
    ],
    "problemMatcher": [],
    "group": {
      "kind": "build",
      "isDefault": true
    }
  }
]
```

^{*4} 当環境に起因しているかは不明です。

Variable (参照値)	Description
<code>\${userHome}</code>	Path of the user's home folder
<code>\${workspaceFolder}</code>	Path of the folder opened in VS Code
<code>\${workspaceFolderBasename}</code>	Name of the folder opened in VS Code without any slashes (/)
<code>\${file}</code>	Currently opened file
<code>\${fileWorkspaceFolder}</code>	Currently opened file's workspace folder
<code>\${relativeFile}</code>	Currently opened file relative to workspaceFolder
<code>\${relativeFileDirname}</code>	Currently opened file's dirname relative to workspaceFolder
<code>\${fileBasename}</code>	Currently opened file's basename
<code>\${fileBasenameNoExtension}</code>	Currently opened file's basename with no file extension
<code>\${fileExtname}</code>	Currently opened file's extension
<code>\${fileDirname}</code>	Currently opened file's folder path
<code>\${fileDirnameBasename}</code>	Currently opened file's folder name
<code>\${cwd}</code>	Task runner's current working directory upon the startup of VS Code
<code>\${lineNumber}</code>	Currently selected line number in the active file

3.5.2 Task 起動のショートカット設定

- **keybindings.json** の編集：

Ctrl+Shift+P ⇨ コマンドパレット ⇨ Preferences: Open Keyboard Shortcuts(JSON)

```
{
  "key": "Alt+F5", // ショートカットキー
  "command": "workbench.action.tasks.runTask",
  "args": "LuaLatex Build", // tasks.json で設定したタスク名
  "when": "editorLangId == latex" // latex ファイルに適用
}
```

3.5.3 Task の起動

Terminal ⇨ Run Build Task

"Alt+F5" // ショートカットキー

3.6 VSCode

- 設定・操作その他の詳細は **VSCode** を参照して下さい。
- ROOT ユーザーでの編集が不可です。

4 TIPS

4.1 PDFViewer

- Linux 系ではデフォルトの PDF ビューワとして **evince** がプリインストールされていることが多いですが、「evince-thumbnail というデーモンが複数実行されたままになり 膨大なリソースを奪う」というバグの問題が確認されていますので、Atril その他のビューワの使用をお勧めします。

\$ apt-get install atril (Debian 系)

- 他のビューワを使用するには、ビューワ名を置き換えます：

```
"command": "... && <viewer name> ...",
```

4.2 連続するハイフン (余録)

- ネット情報では、連続するハイフン (--) を繋がらないで表記するには、”{-}{-}” と記述するとありましたが有効とはなりませんでした。`\hspace{ }`を間に挿むことで解決しました。

以下に幾つかの記法を示します (Lua^AT_EX : ltjsarticle 環境)。

記述	表記	処理内容
--	—	無し (ママ)
{-}{-}	—	{ }で囲む
\textendash	—	TeX コマンド (正常表記)
-\,-	--	空白 : \,
-\hspace{0.4pt}-	--	\hspace{ }