

# Snippets と External Tools(Tasks) の活用 with $\text{\LaTeX}$

[ru\\_museum](#): GitHub)

更新: 2025 年 11 月 24 日

# 目次

1	概要	1
2	作業環境	1
3	Editors	1
3.1	Kate . . . . .	1
3.1.1	Snippets . . . . .	1
3.1.2	External Tools . . . . .	2
3.1.3	Variables 一覧 . . . . .	3
3.2	VSCodium . . . . .	4
3.2.1	Snippets . . . . .	4
3.2.2	Tasks . . . . .	5
3.2.3	Variables 一覧 . . . . .	8
3.3	Pluma . . . . .	10
3.3.1	Snippets . . . . .	10
3.3.2	External Tools . . . . .	11
3.3.3	Variables 一覧 . . . . .	12
4	TIPS	13
4.1	VSCodium : Build の注意点 . . . . .	13

## 1 概要

- ・ 従来から使用して来た **Gedit** における不具合により代替エディタの使用を余儀なくされており、ここでは特に **LaTEX** 文書編集用途の「**Snippets**」「**External Tools(Tasks)**」の活用事例を解説しています。
- ・ その他 **Kate**、**VSCodium**、**KDevelop** 等については以下の解説も参照して下さい。  
⇒ 「**LaTEX** 文書作成における代替エディタの検討.pdf」

## 2 作業環境

GNU/Linux Debian : 6.17.7-2(2025-11-06) x86\_64

Kate : 25.04.3

VSCodium : 1.105.17075

Pluma : 1.28.0-3

- ※ Pluma の Snippets の動作不良は最新バージョンで修正されています。
- ※ KDevelop(4:24.12.3-1) は重大なエラー<sup>\*1</sup> が発生している為に、又 Zed は日本語入力不可により除外しています。
- ※ VSCode は非推奨ですが、処理手順としては VSCodium と共にしていますので参考にして下さい。

## 3 Editors

### 3.1 Kate

#### 3.1.1 Snippets

##### (A) Snippets の有効化

- ・ プラグイン Snippets はデフォルトで組込まれています：  
設定 ⇒ プラグイン ⇒ Snippets

##### (B) Snippet の追加・編集

1. 画面右サイドのツールバーのアイコンで表示・非表示させます。
2. 「Add Snippet」「Edit Snippet」で追加・編集を行います。
3. 挿入後のカーソル位置は、 **\${cursor}** で指示します。

---

<sup>\*1</sup> 起動後ダイアログの点滅表示が繰り替えされるという動作不良が最近発生し使用不可となっています。

【エラー内容】

# kdevelop

kdevplatform.util: Path::init: invalid/unsupported Path encountered: http://clang-tidy  
kdevplatform.shell: Could not load plugin "kdevperforce" , it reported the error: "The variable P4CONFIG is not set. Is perforce installed on the system?" Disabling the plugin now.  
.....(略).....

### 3.1.2 External Tools

#### (A) Tool の新規作成

##### 1. Tool の追加と設定：

Tool の範型から選択します。

設定 ⇒ External Tools ⇒ Add ⇒ Add Tool from Defaults ⇒ Tools ⇒ Run Shell Script

Lualatex による設定例：

項目	入力値
Name	Lualatex Build
Executable	sh
Arguments	-c 'lualatex --shell-escape "%{Document:FileName}" && atril "%{Document:NativePath}"/%{Document:FileBaseName}.pdf"
Input	
Working directory	%{Document:Path}
Mime types	text/x-tex
Save	Current Document
Trigger	After Save
Output	Display in Pane
Editor command	lualatex

- Arguments 欄右端の [ ] (Insert Valiable) をクリックすると Valiable 一覧が表示され入力が出来ます。入力後カーソルを当てる実際値が確認出来ます。
- Mime types はアイコンをクリックすることで選択出来ます。

2. 保存 (Ctrl+S 或いは設定した場合ショートカット 「Alt+F5」<sup>\*2</sup>) でビルドが開始されます。

3. ビルド後に Atril<sup>\*3</sup>で PDF が開かれます。

#### (B) スクリプトの使用

- 添付スクリプト lualatex-build.sh は、ビルドに Biber<sup>\*4</sup>の処理を挿入することを目的としています。

1. (A) Tool の新規作成を参考に Arguments を以下にします：

`-c './lualatex-build1.sh "%{Document:FileName}"'`

※ Tool 側で PDF を開く場合は、以下とします：<sup>\*5</sup>

`-c './lualatex-build1.sh "%{Document:FileName}" && atril "%{Document:NativePath}"/%{Document:FileBaseName}.pdf"`

<sup>\*2</sup> 設定 ⇒ キーボードショートカットを設定 ⇒ アクション名を選択 ⇒ カスタム ⇒ Alt+F5 をキーボードで入力

<sup>\*3</sup> 推奨のビューワー Atril 以外自由に設定出来ます。

<sup>\*4</sup> 文献リスト作成の為のパッケージ。

Biber は変更を反映をさせる為に 2 度のコンパイルを必要とし。生成された.bcf ファイルが存在する場合にスクリプトはコンパイルを再度実行します。

<sup>\*5</sup> スクリプト側でコメントアウトする必要があります。PDFViewer Atril は自由に設定出来ます。

### 3.1.3 Variables 一覧

Variables	参照内容
Document:FileName	File base name without path and suffix of the current document.
Document:FileExtension	File extension of the current document.
Document:FilePath	Full path of the current document including the file name
Document:Text	Contents of the current document.
Document:Path	Full path of the current document excluding the file name.
Document:NativeFilePath	Full document path including file name, with native path separator(backslash on Windows).
Document:NativePath	Full document path excluding file name, with native path separator(backslash on Windows).
Document:Cursor:Line	Line number of the text cursor position in current document (starts with 0).
Document:Cursor:Column	Column number of the text cursor position in current document (starts with 0).
Document:Cursor:XPos	X component in global screen coordinates of the cursor position.
Document:Cursor:YPos	Y component in global screen coordinates of the cursor position.
Document:Selection:Text	Text selection of the current document.
Document:Selection:StartLine	Start line of selected text of the current document.
Document:Selection:StartColumn	Start column of selected text of the current document.
Document:Selection:EndLine	End line of selected text of the current document.
Document:Selection:EndColumn	End column of selected text of the current document.
Document:RowCount	Number of rows of the current document.
Document:Variable:<variable>	Expand arbitrary document variables.
Date:Locale	The current date in current locale format.
Date:ISO	The current date (ISO).
Date:<value>	The current date (QDate formatstring).
Time:Locale	The current time in current locale format.
Time:ISO	The current time (ISO).
Time:<value>	The current time (QTime formatstring).
ENV:<value>	Access to environment variables.
JS:<expression>	Evaluate simple JavaScript statements.
PercentEncoded:<text>	Percent encoded text.
UUID	Generate a new UUID.

## 3.2 VSCodium

VSCodium は MS VSCode から分岐したオープンソース版で、データ収集などのセキュチティ面を除き全ては機能的に同一ですので、以下に掲載の内容は操作も含め VSCode でも共通です。

### 3.2.1 Snippets

- Extension にも幾つか Snippets 用のものがありますが、自作も可能です。
- 作業に伴い最適化したカスタマイズを行います。

#### (A) latex.json の作成

- Ctrl+Shift+P でコマンドパレットを開く。
- snippet と入力。
- Snippets: Configure Snippets を選択。
- latex を選択。
- ./config/Code/User/snippets/latex.json にファイルが生成されます。

※ 手動で設置することも出来ます（参照：「(B) 添付 latex.json の使用」）。

#### (B) 添付 latex.json の使用

- 添付 snippets-template-latex.json を latex.json と変更します。
- ./config/Code/User/snippets/latex.json を置きます。  
或いは必要部分をコピーします（自由に編集可能）。
- 追加された snippet は、語頭に「□」印が付されて候補に列挙されます（例：□ sample）。

#### 作成・編集上の注意点

コード	表記例
\$0,\$1,\$2	挿入時のカーソル位置指定 (\$0 は最終位置)
\hspace	\hspace
改行 \\	改行\\\\\\ ( \\ = \\\\" で表現)

#### (C) 日時 snippets

- date

2025 年 11 月 01 日 (土) 07:34:02

```
// 記述例：  
"$CURRENT_YEAR 年$CURRENT_MONTH 月$CURRENT_DATE 曰 ($CURRENT_DAY_NAME_SHORT)  
$CURRENT_HOUR:$CURRENT_MINUTE:$CURRENT_SECOND"
```

※設定値は以下の variableslist で確認出来ます。

- **variableslist**

表記	Variables	表記内容
2025	CURRENT_YEAR	current year
25	CURRENT_YEAR_SHORT	current year's last two digits
11	CURRENT_MONTH	month as two digits (example '02')
11月	CURRENT_MONTH_NAME	full name of the month (example 'July')
11月	CURRENT_MONTH_NAME_SHORT	short name of the month (example 'Jul')
01	CURRENT_DATE	day of the month as two digits (example '08')
土曜日	CURRENT_DAY_NAME	name of day (example 'Monday')
土	CURRENT_DAY_NAME_SHORT	short name of the day (example 'Mon')
08	CURRENT_HOUR	current hour in 24-hour clock format
07	CURRENT_MINUTE	current minute as two digits
22	CURRENT_SECOND	current second as two digits
1761952042	CURRENT_SECONDS_UNIX	number of seconds since the Unix epoch
+09:00	CURRENT_TIMEZONE_OFFSET	current UTC time zone offset as +HH:MM or -HH:MM (example -07:00)

※ VSCodium を日本語表示にしていると日本語での表記となります。

### 3.2.2 Tasks

#### (A) tasks.json の基本的使用

- 基本的 Build 例 :

```
{
  "label": "Lualatex Build",
  "type": "shell",
  "command": "lualatex",
  "args": [
    "--shell-escape", "--output-directory='${fileDirname}' '${file};" ,
    "atril", "'${fileDirname}/${fileBasenameNoExtension}.pdf'"
  ],
  "problemMatcher": [],
  "group": {
    "kind": "build",
    "isDefault": true
  }
}
```

1. Ctrl+Shift+B でコマンドパレットを開きビルド名 (Lualatex Build) を選択し実行します。
2. lualatex によるビルト後に PDF ファイル (atril) が開かれます。

## (B) スクリプトの使用

- スクリプトは、主であるビルド処理に追加処理を挿むことを目的としています。
- このスクリプトは、途中に Biber<sup>\*6</sup>の処理を挿入しています。

```
#!/bin/sh

#####
# LUALATEX BUILD 処理スクリプト
# lualatex-build.sh *.tex
#####

FILENAME="$1" # 引数から実行ファイル取得
LUA_CMD="lualatex -shell-escape"

# 生成されるファイル
FILE_PDF=`basename "$FILENAME" .tex`.pdf"
FILE_BCF=`basename "$FILENAME" .tex`.bcf"

# LUALATEX の実行
if [ -e $1 ]; then

    echo "\n=====$1 の処理を開始します:"
    $LUA_CMD $FILENAME
    echo "処理は正常に終了しました。PDF ファイルが開かれます:\n"

else

    echo "-----"
    echo " ファイル名エラー : [" $1 "]"
    echo " 処理に失敗しました。"
    echo "  ⇒ ファイルが存在するかを確認して下さい。"
    echo "-----\n"
    return

fi

# BIBER .BCF が存在する場合 :
if [ -e $FILE_BCF ]; then

    echo "BCF File のコンパイルを開始します -----"
    biber $FILE_BCF
    echo "BCF File のコンパイルを終了しました -----"

    echo "処理を反映させる為、再度のコンパイルを行います -----"
    $LUA_CMD $FILENAME
    echo "処理は反映されました -----"

fi

# PDF を開く : TASK 側で開く場合はコメントアウトします
if [ -e $FILE_PDF ]; then

    atril $FILE_PDF # Atril 以外も利用可能
    echo "$FILE_PDF を開きました"

fi

exit 0
```

- 作業フォルダに lualatex-build.sh を設置します<sup>\*7</sup>。
- \$ chmod 744 lualatex-build.sh // 実行権限を与えます<sup>\*8</sup>

\*6 2ページ \*3 を参照

\*7 スクリプトは、Snippet からも作成可能としています。言語モード Latex 環境で buildscript で生成されます。  
作業フォルダごとの設定が容易になります。

\*8 直接実行する場合:\$ ./lualatex-build.sh <filename>

### 3. tasks.json への登録：

(A) スクリプト側で PDF を開く場合：

```
{
    "label": "lualatexbuild", // 名前は自由に付けられます
    "type": "shell",
    "command": "${fileDirname}/lualatex-build.sh ${file}",
    "problemMatcher": [],
    "group": {
        "kind": "build",
        "isDefault": true
    }
},
```

(B) Task 側で PDF を開く場合<sup>\*9</sup>：

```
{
    "label": "lualatexbuild", // 名前は自由に付けられます
    "type": "shell",
    "command": "${fileDirname}/lualatex-build.sh ${file};",
        "atril", "'${fileDirname}/${fileBasenameNoExtension}.pdf'"
    "problemMatcher": [],
    "group": {
        "kind": "build",
        "isDefault": true
    }
},
```

4. Ctrl+Shift+B でコマンドパレットを開きビルド名を選択するとスクリプトが実行されます。

(C) スクリプトの新規作成

- 添付の Snippet(latex.json) より作成することで、作業フォルダごとの設置が容易になります。

1. ./config/Code/User/snippets/latex.json を置きます。

或いは、既存の latex.json があれば当該部分を追加します。

2. 「新しいテキスト ファイル」を作成し右下にある「言語モードの選択」で「LaTeX」を選択します。

3. "b" を入力して表示された候補  buildscript を選択しスクリプトを読み込みます。

4. ファイルを lualatex-build.sh として保存し、アクセス権の「実行の許可」を与えます：

```
$ chmod 744 lualatex-build.sh
```

5. スクリプトの実行方法は「(B) スクリプトの使用 3. tasks.json への登録」を参照して下さい。

---

<sup>\*9</sup> スクリプトの「PDF を開く」部分をコメントアウトする必要があります。

### 3.2.3 Variables 一覧

#### (A) 編集上の VARIABLES

Variables	表記内容
TM_SELECTED_TEXT	The currently selected text or the empty string
TM_CURRENT_LINE	The contents of the current line
TM_CURRENT_WORD	The contents of the word under cursor or the empty string
TM_LINE_INDEX	The zero-index based line number
TM_LINE_NUMBER	The one-index based line number
TM_FILENAME	The filename of the current document
TM_FILENAME_BASE	The filename of the current document without its extensions
TM_DIRECTORY	The directory of the current document
TM_FILEPATH	The full file path of the current document
RELATIVE_FILEPATH	The relative (to the opened workspace or folder) file path of the current document
CLIPBOARD	The contents of your clipboard
WORKSPACE_NAME	The name of the opened workspace or folder
WORKSPACE_FOLDER	The path of the opened workspace or folder
CURSOR_INDEX	The zero-index based cursor number
CURSOR_NUMBER	The one-index based cursor number

#### (B) 日時関連の VARIABLES

Variables	表記	表記内容
CURRENT_YEAR	2025	The current year
CURRENT_YEAR_SHORT	25	The current year's last two digits
CURRENT_MONTH	02	The month as two digits
CURRENT_MONTH_NAME	October	The full name of the month
CURRENT_MONTH_NAME_SHORT	Oct	The short name of the month
CURRENT_DATE	08	The day of the month as two digits
CURRENT_DAY_NAME	Friday	The name of day
CURRENT_DAY_NAME_SHORT	Fri	The short name of the day
CURRENT_HOUR	11	The current hour in 24-hour clock format
CURRENT_MINUTE	03	The current minute as two digits
CURRENT_SECOND	02	The current second as two digits
CURRENT_SECONDS_UNIX	1761878012	The number of seconds since the Unix epoch
CURRENT_TIMEZONE_OFFSET	+09:00	The current UTC time zone offset as +HH:MM or -HH:MM

### (C) その他の VARIABLES

Variables	表記内容
RANDOM	6 random Base-10 digits
RANDOM_HEX	6 random Base-16 digits
UUID	A Version 4 UUID
BLOCK_COMMENT_START	Example output: in PHP /* or in HTML <!--
BLOCK_COMMENT_END	Example output: in PHP */ or in HTML -->
LINE_COMMENT	Example output: in PHP //

### 3.3 Pluma

#### 3.3.1 Snippets

- Snippets<sup>\*10</sup>の設定及び編集方法は、Gedit のページ<sup>\*11</sup>を参照して下さい。殆どの組込変数はPREFIXをPLUMAに変更することで利用可能です。  
例：\$GEDIT\_SELECTED\_TEXT ⇌ \$PLUMA\_SELECTED\_TEXT
- 表示結果のカーソル位置：\$0(最終位置)、\$1、\$2 ...
- 改行コード：\\ ⇌ \\\\で表記します。
- Pythonの利用：  
`$< python code >`

時刻表示例：2025年11月22日(土)22時04分24秒

```
$<1:  
import time;  
# returnで文字列を返します  
return time.strftime("%Y年%m月%d日(%a)%H時%M分%S秒")  
>
```

書式	表記	内容
%d	22	0埋めした10進数で表記した月中の日にち
%m	11	0埋めした10進数で表記した月
%y	25	0埋めした10進数で表記した西暦の下2桁
%Y	2025	0埋めした10進数で表記した西暦4桁
%H	22	0埋めした10進数で表記した時(24時間表記)
%I	10	0埋めした10進数で表記した時(12時間表記)
%M	04	0埋めした10進数で表記した分
%S	24	0埋めした10進数で表記した秒
%A	土曜日	ロケールの曜日名
%a	土	ロケールの曜日名(短縮形)
%B	11月	ロケールの月名
%b	11月	ロケールの月名(短縮形)
%j	326	0埋めした10進数で表記した年中の日にち(正月が'001')
%U	46	0埋めした10進数で表記した年中の週番号(週の始まりは日曜日)
%W	46	0埋めした10進数で表記した年中の週番号(週の始まりは月曜日)

<sup>\*10</sup> Snippetsの不具合は最近のアップデートで解消されています(2025-11-24)。

<sup>\*11</sup> [Apps/Gedit/Plugins/Snippets // GNOME](https://wiki.gnome.org/Apps/Gedit/Plugins/Snippets)

<https://wiki.gnome.org/Apps/Gedit/Plugins/Snippets>

### 3.3.2 External Tools

- External Tools の設定及び編集方法は、Gedit のページ<sup>\*12</sup>を参照して下さい。殆どの組込変数は PREFIX を PLUMA に変更することで利用可能です。  
例：\$GEDIT\_CURRENT\_DOCUMENT\_NAME ⇌ \$PLUMA\_CURRENT\_DOCUMENT\_NAME
- 外部ツールマネージャ (Manage External Tools) におけるスクリプト使用例：

(1) 「ツール」で新規作成「LuaLatex Build」として以下を入力します。

設定項目	入力値
ショートカットキー	<Alt>F5
保存	現在の文書
入力	現在の文書
出力	下端のペインを表示
適用範囲	ローカルファイルのみ

(2) 「編集」欄にスクリプトを入力します。

```
#!/bin/sh

FILENAME="$1" # 引数から実行ファイル取得
LUA_CMD="lualatex -shell-escape"

# 生成されるファイル
FILE_PDF=`basename "$FILENAME" .tex`.pdf"
FILE_BCF=`basename "$FILENAME" .tex`.bcf"

# LUALATEX の実行
if [ -e $1 ]; then
    echo "\n=====$1 の処理を開始します:"
    $LUA_CMD $FILENAME
    echo "処理は正常に終了しました。PDF ファイルが開かれます:\n"
else
    echo "-----"
    echo " ファイル名エラー:[ \"$1 \" ]"
    echo " 処理に失敗しました。"
    echo " ⇒ ファイルが存在するかを確認して下さい。"
    echo "-----\n"
    return
fi

# BIBER .BCF が存在する場合：
if [ -e $FILE_BCF ]; then
    echo "BCF File のコンパイルを開始します -----"
    biber $FILE_BCF
    echo "BCF File のコンパイルを終了しました -----"
    echo "処理を反映させる為、再度のコンパイルを行います -----"
    $LUA_CMD $FILENAME
    echo "処理は反映されました -----"
fi
```

\*12 [Apps/Gedit/Plugins/ExternalTools](https://wiki.gnome.org/Apps/Gedit/Plugins/ExternalTools) // GNOME  
<https://wiki.gnome.org/Apps/Gedit/Plugins/ExternalTools>

```

# PDFを開く：IDFViewer (atril) は自由に変更可能です
if [ -e $FILE_PDF ]; then
    atril $FILE_PDF # Atril 以外も利用可能
    echo "$FILE_PDF を開きました"
fi
exit 0

```

(3) ショートカットキー「<Alt>+F5」<sup>\*13</sup> で文書は保存され、PDF ファイルが開かれます。

### 3.3.3 Variables 一覧

Variables	表記
\$PLUMA_CURRENT_DOCUMENT_DIR	/your dirname
\$PLUMA_CURRENT_DOCUMENT_NAME	test.tex
\$PLUMA_CURRENT_DOCUMENT_PATH	/your dirname/test.tex
\$PLUMA_CURRENT_DOCUMENT_SCHEME	file
\$PLUMA_CURRENT_DOCUMENT_TYPE	text/x-tex
\$PLUMA_CURRENT_DOCUMENT_URI	file:///your dirname/test.tex
\$PLUMA_CURRENT_LINE	your words
\$PLUMA_SELECTED_TEXT	your selected text
\$PLUMA_CURRENT_LINE_NUMBER	000
\$PLUMA_CURRENT_WORD	your-word
\$PLUMA_DOCUMENTS_PATH	/your dirname/test1.tex /your dirname/test2.tex
\$PLUMA_DOCUMENTS_URI	file:///your dirname/test1.tex file:///your dirname/test2.tex

---

<sup>\*13</sup> ショートカットキーは自由に設定可能です。

多くのディストリビューションで同梱されている PDFViewer の Evince は不具合が確認されているため非推奨です。

## 4 TIPS

### 4.1 VSCodium : Build の注意点

- LuaLaTeX では SVG などの画像読み込みには Inkscape が使われていますが、作業フォルダを正式にメニューの「ファイル (File)」から開かないと Inkscape からファイルへの PATH が正しく通らずエラーとなります。
- PDF ファイルが開かれ Task is running の状態の時には、再度「Alt+F5」での起動は一度 Command Pallet を介して実行されます。
- 添付シェルスクリプトの lualatex-build.sh は、コンパイル時に自由に「追加の処理」を行うことが出来ます。