

# L<sup>A</sup>T<sub>E</sub>X 入門 with VSCode(仮)

2K

2022 年 12 月 3 日

## 1 はじめに

これは VSCode で最高の L<sup>A</sup>T<sub>E</sub>X の環境を整えるための備忘録です。私が記憶喪失になった場合に備え、かなり丁寧に説明してあります。LaTeX、VSCode について知らなくても、ここに書いてある手順に従えば分かるはずです。以下注意点です。

まず、この資料は完璧でないので他の資料と見比べながら進めてもらった方が安全です。L<sup>A</sup>T<sub>E</sub>X に関することならとりあえず <https://texwiki.texjp.org/> を見ておけば大丈夫でしょう。説明が雑な所も多いので分からなかったらネットで調べてください。

また、あらかじめファイル管理システム (Windows の場合はエクスプローラー) の設定をファイル名拡張子を表示するように変更しておくといいです。もし「ファイル」、「フォルダ」、「ディレクトリ」、「拡張子」という単語になじみがない場合は調べて理解しておきましょう。

なお、OS は Windows10 を想定しています。他のバージョンの Windows だと表示が少し違うようですが、やることは余り変わりません。友人によると Windows11 でも問題なくできたそうです。macOS、Linux などの場合は大きく異なる点が出てくるので十分注意してください。

最後に、これはあくまで私のための備忘録なので、これを参考にして何か問題が起こっても全て自己責任でお願いします。

## 2 T<sub>E</sub>XLive

### 2.1 T<sub>E</sub>XLive のインストール

まずは T<sub>E</sub>XLive をパソコンにインストールします。T<sub>E</sub>XLive とは、簡単に言うと L<sup>A</sup>T<sub>E</sub>X を動かすためのプログラムとかを全部まとめたものです。詳しくは <https://texwiki.texjp.org/?TeX%20Live> を参照してください。

Windows の場合は <https://texwiki.texjp.org/?TeX%20Live%2FWindows> からインストーラをダウンロードできます。基本的にはネットワークインストーラを用いる方法が良いと思います。他の OS の場合は対応するページを見てください。

この T<sub>E</sub>XLive、容量が 3.4GB 程と非常に重たいです。パソコンの空き容量は 10GB 以上あった方が安心です。また、インストールに 1、2 時間かかるので、通信環境の良い場所で時間がある時にやりましょう。

## 2.2 環境変数の確認

インストールが終わったら環境変数の確認をします。これは万全を期すための確認なので、よく分からない場合は 2.3 節に飛んでもらっても構いません。

Windows の場合、タスクバーの検索ボックスに「システム環境変数」と入力してください。すると図 1 のような画面が表示されるので、「システム環境変数の編集」を開いてください。「環境変数の編集」というものもありますが、これではないので気を付けてください。図 2 のようなウィンドウが開いたら右下の「環境変数」をクリックします。さらに図 3 のようなウィンドウが開くので、「システム環境変数」の枠内の「Path」をダブルクリックしてください。そこに

```
%SystemRoot%\system32
```

と

```
%SystemRoot%
```

の 2 つが含まれていれば OK です。なければ「新規」を押して追加してください。

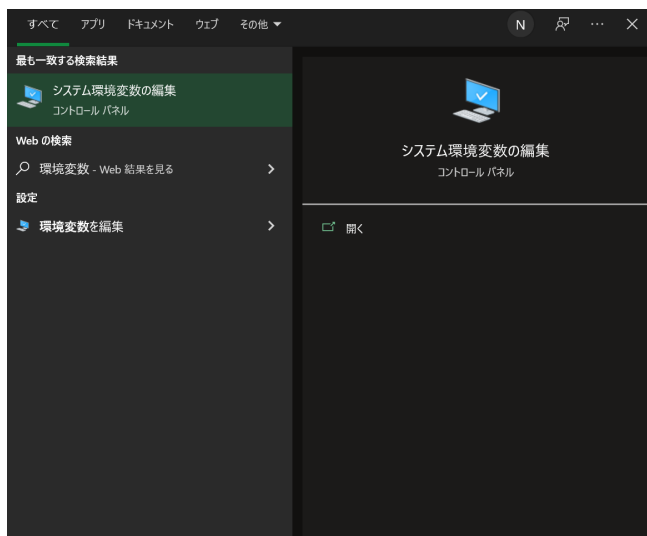


図 1



図 2

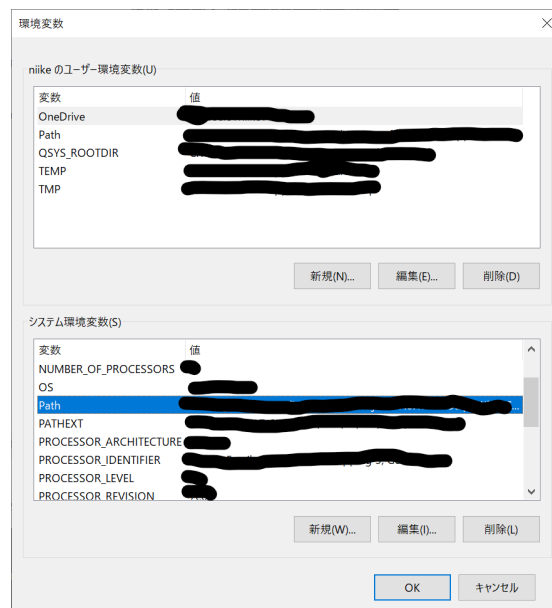


図 3

## 2.3 TeXLive のテスト

次に、 $\text{\LaTeX}$  を実際に書いて TeXLive が正常にインストールされていることを確認します。ここでは TeXworks というエディタを使います。タスクバーの検索ボックスに「texworks editor」と入力するか、スタートメニューの「TeX Live 2021」→「TeXworks editor」から起動してください。エディタが表示されたら、次のコード 1 を貼り付けてください。

ソースコード 1: TeXLive のテストコード

```
\documentclass{article}
\begin{document}
\begin{quote}
  \begin{verbatim*}
    Hello \LaTeX !\
    こんにちは \LaTeX !
  \end{verbatim*}
\end{quote}
\end{document}
```

左上の緑の三角のボタンを押すとファイルの保存場所を聞かれるので、適当な場所に保存してください。ビルドが始まり、図 4 のように表示されたら成功です。上手く行かない場合はまず入力内容が正しいか確かめ、それでもダメなら <https://texwiki.texjp.org/?Microsoft%20Windows> を見て指示に従ってください。

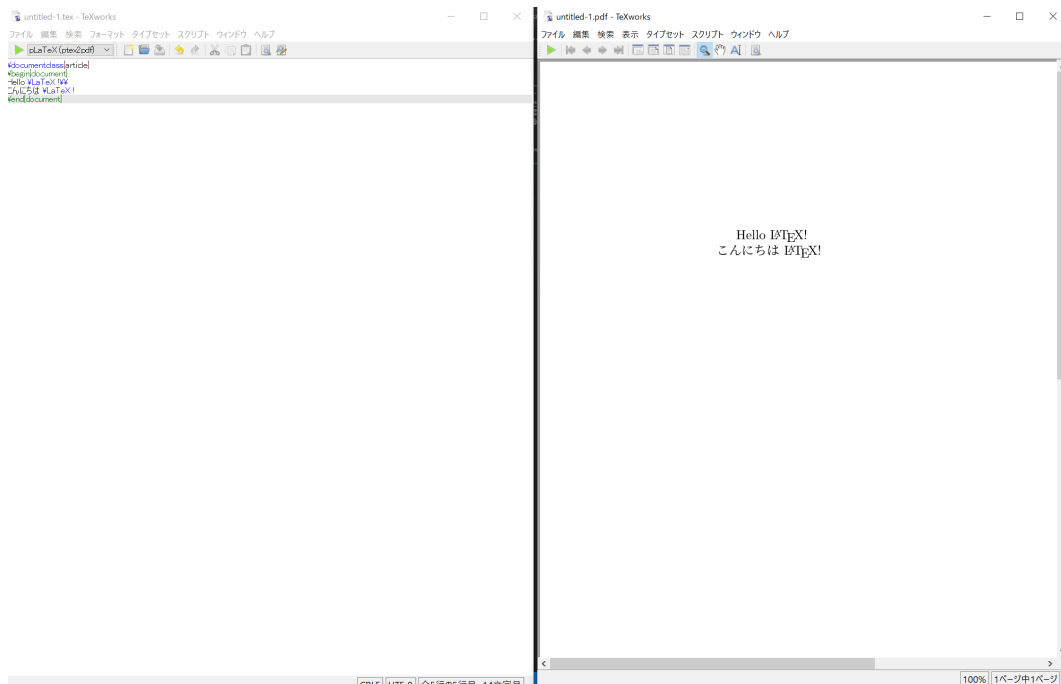


図 4

L<sup>A</sup>T<sub>E</sub>X には様々なビルドの方式があります。先程の緑の実行ボタンの右に書いてあるのが現在使っている方式の名前です。クリックすると図 5 のようになってどの方式を使うかを選択することができます。今回は LuaL<sup>A</sup>T<sub>E</sub>X という方式を使うので、「LuaLaTeX」を選択して再度実行ボタンを押し、正しくビルドされることを確認してください。問題がなければアプリを終了してもらって構いません。

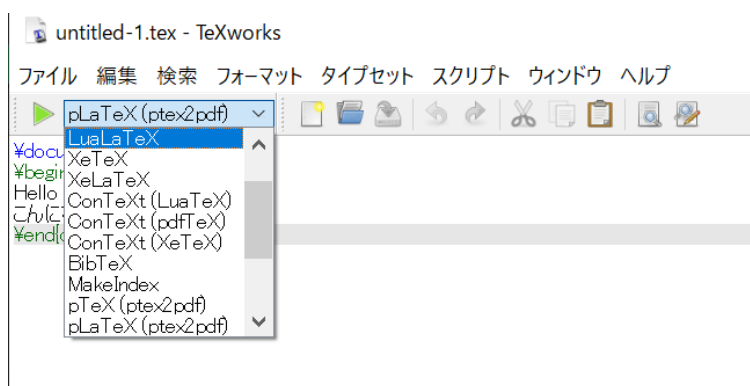


図 5

これで L<sup>A</sup>T<sub>E</sub>X 自体の設定は終わりです。この先を書いてあることをしなくても、T<sub>E</sub>XLive を使えばレポートを書くことはできます。ここからは、VSCode というエディタを使って L<sup>A</sup>T<sub>E</sub>X をよりスマートに編集する方法について説明します。

## 3 VSCode

### 3.1 VSCode のインストール

3 節では VSCode のインストール方法と基本設定について説明します。VSCode を既に使っている場合は 4 節に飛んでもらっても構いません。

VSCode(Visual Studio Code) は、Microsoft が提供する無料のコードエディターです。<https://code.visualstudio.com/download> からダウンロードできます。Windows の場合は Windows ロゴの真下にあるボタンを押して exe ファイルを適当な場所にダウンロードし、開いてください。なお、その時に英語で書かれたページに遷移するので、興味がある場合は覗いてみましょう。初心者用の解説動画があったりします。

exe ファイルを開いたらインストーラーが起動します。基本的には何もいじらずに進めてもらった方が良いのですが、一か所だけして欲しい設定があります。「追加タスクの選択」という画面が表示されたら、特にこだわりがなければ図 6 のようにチェックを入れてください。

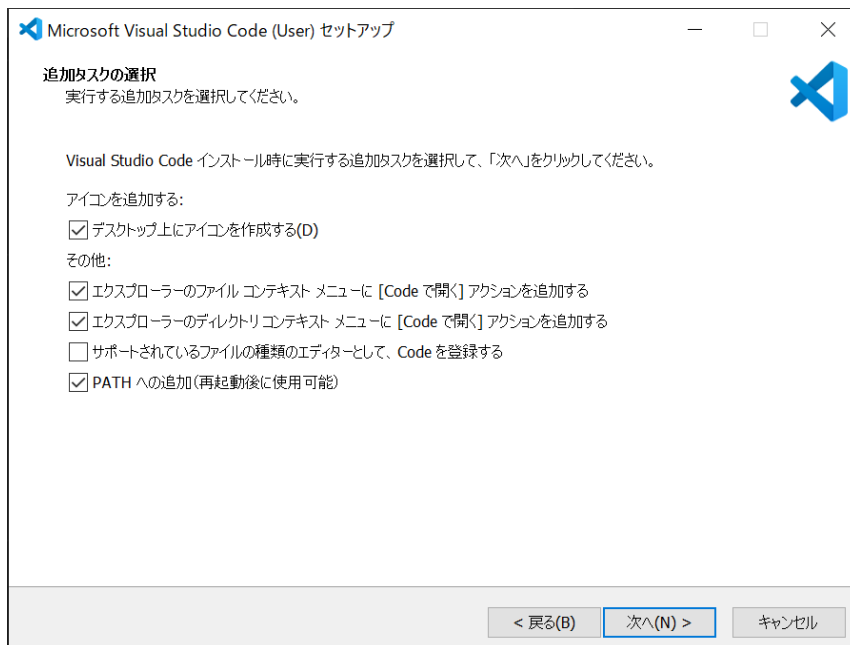


図 6

各項目について一応説明しておきます。気になる人は読んでください。

デスクトップ上にアイコンを作成する

そのままの意味です。

エクスプローラーのファイルコンテキストメニューに [Code で開く] アクションを追加する

エクスプローラーでファイルの上で右クリックすると図7のように「Code で開く」という選択肢が現れるようになります。これをクリックすれば VSCode 上でファイルを開けます。

エクスプローラーのディレクトリコンテキストメニューに [Code で開く] アクションを追加する

上のことがフォルダに対してもできるようになります。これはよく使うので必ずチェックを付けましょう。

サポートされているファイルの種類のエディターとして、Code を登録する

VSCode で開けるファイルをダブルクリックすると、直接 VSCode で開くようになります。VSCode を既定のエディタとして利用するならチェックを付けても良いですが、別に付けなくても構いません。迷うようならチェックははずして良いでしょう。

PATH への追加 (再起動後に使用可能)

コマンドプロンプトで `code` と打つと VSCode を起動できるようになります。また、`code_<ファイル名>` でファイルを、`code_<フォルダ名>` でフォルダを開けるようになります。よく分からなくてもとりあえずチェックを入れておいて大丈夫です。

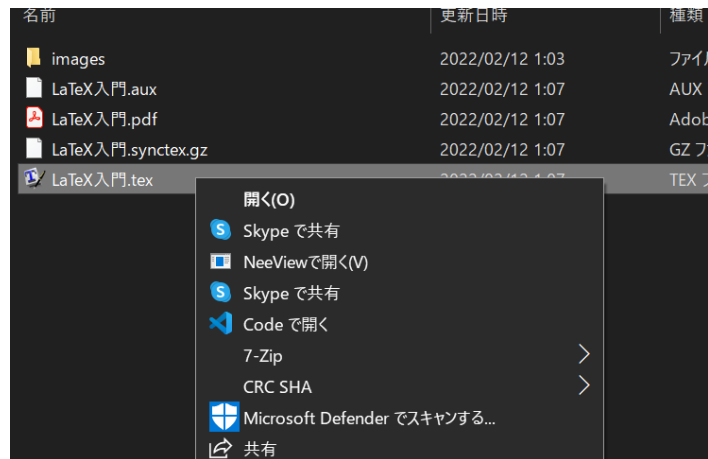


図 7

チェックの入れ忘れ、間違いなどがあり後で訂正したい場合は、VSCode をアンインストールすることなく、もう一度インストーラを開いて設定し直してください (未確認)。

### 3.2 VSCode の日本語化

インストールが終わったら自動的に VSCode が起動します。起動しない場合、デスクトップのアイコンをダブルクリックするか、タスクバーの検索欄に「vscode」と入力するなどして立ち上げてください。

次に、VSCode の日本語化のための拡張機能を入れます (もちろん入れなくても構いません)。左端にあるアイコンの列の中から、田んぼの「田」の右上が切り離された感じのアイコンをクリックしてください。すると検索欄が出てくるので、「japanese」と入力して一番上に出てくる「Japanese Language Pack for Visual Studio Code」をインストールしてください (図 8)。



図 8

すると画面の右下に英語が書かれたボタンが出てくるので、リロードして日本語化を有効にするボタンを押してください。画面に日本語が表示されれば OK です。上手く行かない場合は、図 8 の画面に戻ってそこに書いてある「使用法」に従ってください。

## 4 $\text{\LaTeX}$ with VSCode

### 4.1 拡張機能のインストールと settings.json の編集

次は VSCode 上で  $\text{\LaTeX}$  を動かす準備をします。まずは必要な拡張機能をインストールしましょう。拡張機能のボタンを押して「latex」と検索し、「LaTeX Workshop」をインストールしてください (図 9)。



図 9

次に、settings.json に LaTeX の設定を追加します。settings.json とは VSCode のユーザー設定が記述されたファイルです。このファイルを編集することで VSCode を自由にカスタマイズすることができます。

Ctrl+Shift+P を押すと画面上部に検索欄 (コマンドパレット) が出てくるので、そこに「設定」と入力しましょう。すると図 10 のように設定関係の操作の候補が出てくるので、その中から「基本設定: 設定 (JSON) を開く」を選択します。「既定の設定 (JSON) を開く」とは別物なので注意してください。



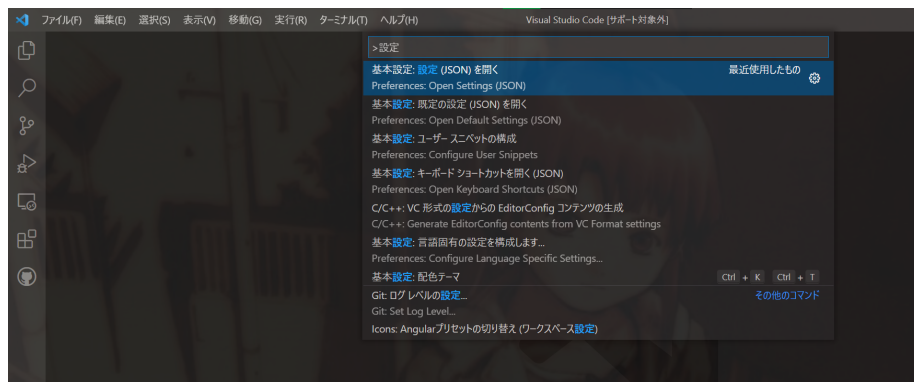


図 10

すると、settings.json というファイルが開きます。まだ何も設定をしていない場合、空欄か中身がない中括弧だけが表示されると思います。

このファイルでは次のように設定を記述します。

```
{
  "<設定 1 の名前>": <設定 1 の内容>,
  "<設定 2 の名前>": <設定 2 の内容>,
  ...
}
```

一番外側の中括弧と、各設定の間のカンマを忘れないように注意してください。詳しくは json 形式についてネットで調べてください。

settings.json が空欄の場合は、ソースコード 2 を貼り付けて挙動を確認しましょう。これには VSCode の基本的な設定が記述されています。VSCode 上でカーソルを重ねると各設定についての説明が読めます。

#### ソースコード 2: settings.json の基本的な例

```
{
  "editor.wordWrap": "on",
  "editor.fontSize": 15,
  "editor.renderWhitespace": "all",
  "editor.suggestSelection": "recentlyUsedByPrefix",
  "editor.guides.bracketPairs": true,
  "editor.bracketPairColorization.enabled": true,
  "files.autoSave": "afterDelay", // 自動保存ON
  "files.autoSaveDelay": 1500, // 編集の1500ミリ秒後に自動保存
  "workbench.iconTheme": "vscode-icons",
  "vsicons.dontShowNewVersionMessage": true,
  "explorer.confirmDragAndDrop": false,
```

```

"workbench.editor.untitled.hint": "hidden",
"files.exclude": {
  // VSCodeで以下の拡張子のファイルを表示させないようにする
  "**.aux": true,
  "**.synctex.gz": true,
  "**/.classpath": true,
  "**/.factorypath": true,
  "**/.project": true,
  "**/.settings": true
},
// ここに最初と最後の中括弧を削除したソースコード3を貼り付けてください
}

```

次に、settings.json に L<sup>A</sup>T<sub>E</sub>X を動かすための設定を追加します。

ソースコード 2 を先程貼り付けた場合は、ソースコード 3 の最初と最後の中括弧を削除して、コメントで指定されている場所に貼り付けてください。そうでない場合も json 形式に従って適宜追加してください。

settings.json 上でカーソルを重ねると各設定についての説明が表示されます。詳細は <https://texwiki.texjp.org/?Visual%20Studio%20Code%20LaTeX> などで調べてください。

### ソースコード 3: settings.json の L<sup>A</sup>T<sub>E</sub>X の設定

```

{
  "latex-workshop.latex.tools": [
    {
      "name": "lualatex",
      "command": "lualatex",
      "args": [
        "--cmdx",
        "-file-line-error",
        "-interaction=nonstopmode",
        "-halt-on-error",
        "-synctex=1",
        "%DOC%"
      ],
      "env": {}
    }
  ],
  "latex-workshop.latex.recipes": [
    // 自動コンパイルでは一番上のレシピ(ここではlualatex1)が実行されます
    {
      "name": "lualatex1",
      "tools": [
        "lualatex"
      ]
    }
  ]
}

```

```

    ]
  }
],
"editor.wordSeparators": ". / \\ ( ) \\' - : , . ; < > ~ ! @ # $ % ^ & * | + = [ ] { } ` ~ ? 、 。 「 」 【 】 『 』 ( ) { } 《 》 < > ! ? ~ て に を は が の と も へ で や ",
"latex-workshop.latex.clean.fileTypes": [
  // "*.aux", // 参照を使うときはコメントアウトする
  "*.bbl",
  "*.blg",
  "*.idx",
  "*.ind",
  "*.lof",
  "*.lot",
  "*.out",
  "*.toc", // 目次を作成する時はコメントアウトする
  "*.acn",
  "*.acr",
  "*.alg",
  "*.glg",
  "*.glo",
  "*.gls",
  "*.ist",
  "*.fls",
  "*.log",
  "*.fdb_latexmk",
  // "*.synctex.gz", // synctexを使うときはコメントアウトする
  // for Beamer files
  "_minted*",
  "*.nav",
  "*.snm",
  "*.vrb"
],
"latex-workshop.latex.autoClean.run": "onBuilt",
"latex-workshop.view.pdf.viewer": "tab"
}

```

## 4.2 VSCodeで $\text{\LaTeX}$ のテスト

以上で基本的な設定は終わりです。早速 VSCode で  $\text{\LaTeX}$  を動かしてみましょう。エクスプローラーで適当な場所にフォルダを作成し、それを右クリックして「Codeで開く」を選択して下さい。VSCode が起動します。左上に開いているフォルダの名前が表示されるので、それをクリックして「新しいファイル」を選択して下さい (図 11)。適当な名前 (test.tex など) を付けて Enter キーを

押すとファイルが作成されます。



図 11

作成したファイルにソースコード 4 を貼り付け、右上にある緑の三角のボタンを押してください。

#### ソースコード 4: VSCode のテストコード

```
\documentclass[11pt,a4paper]{bxjsarticle}
\usepackage{luatexja}

\begin{document}
Hello \LaTeXe{} !!
こんにちは \LaTeXe{} !
\end{document}
```

上手くビルドされると左下にチェックマークが表示されます。実行ボタンの右隣のボタンを押すと、画面の右側に pdf ファイルが表示されます (図 12)。

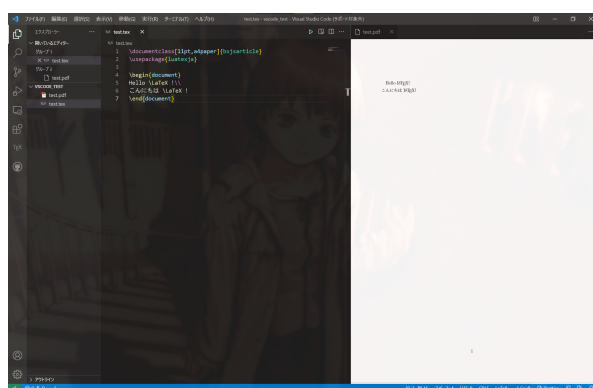


図 12: VSCode でのビルド

pdf ファイルを Ctrl+左クリックすると、tex ファイルの対応する箇所に飛びます。逆に、tex ファイルで Ctrl+Alt+J を押すと pdf ファイルに飛びます (SyncTeX の機能)。

settings.json にソースコード 2 を貼り付けた場合、VSCode でファイルを編集すると自動で保存が行われます。また、4.1 節でインストールした拡張機能「LaTeX Workshop」の既定の設定ではファイルを保存する度に tex ファイルのビルドが行われることになっています。これらの組み合わせにより、tex ファイルを書き換えると実行ボタンを押さなくても自動で pdf ファイルが更新されることになります。自動保存、自動ビルドが嫌な場合は各自で設定を変更してください。(ref などの相互参照を使う場合、2 回ビルドが必要になります。詳しくは調べてください。)

## 5 $\text{\LaTeX}$ の執筆

### 5.1 プリアンブルについて

書けたら書く

### 5.2 スニペットの活用

書けたら書く

### 5.3 Git の活用

書けたら書く

## 6 おまけ: VSCode の便利機能

### 6.1 コマンドパレットについて

VSCode で `Ctrl+Shift+P` を押すと、画面の上部にコマンドパレットが表示されます。ここにやりたいことを入力するとたいていの事ができます。例えば「ファイル」と入力すれば図 13 のようにファイルに関する操作の一覧が表示されます。

VSCode を再起動したいなら「reload」と入力しましょう。図 14 のような画面が開き、上下キーで操作を選択できます。「開発者: ウィンドウの再読み込み」の上で Enter キーを押すかクリックすれば、VSCode を再起動できます (正確には再起動ではなく再読み込み?)。このように、コマンドパレットを使えば直感的な操作を行うことが可能となります。

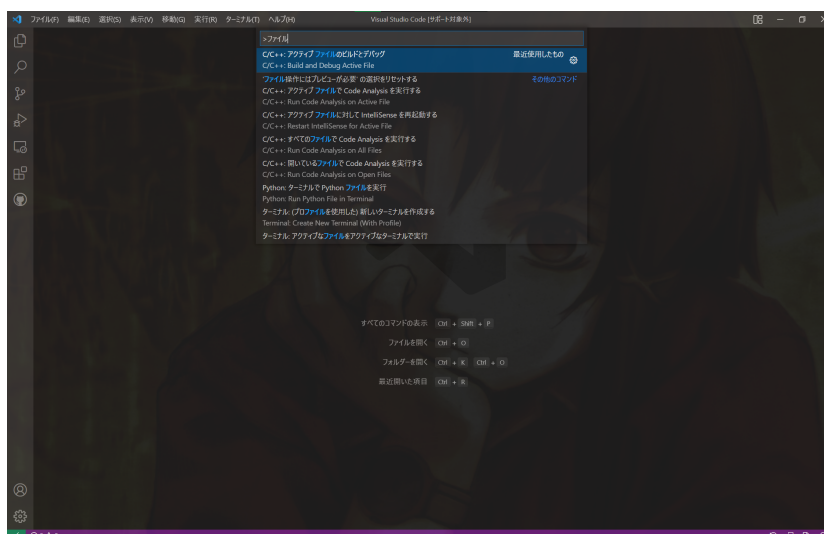


図 13

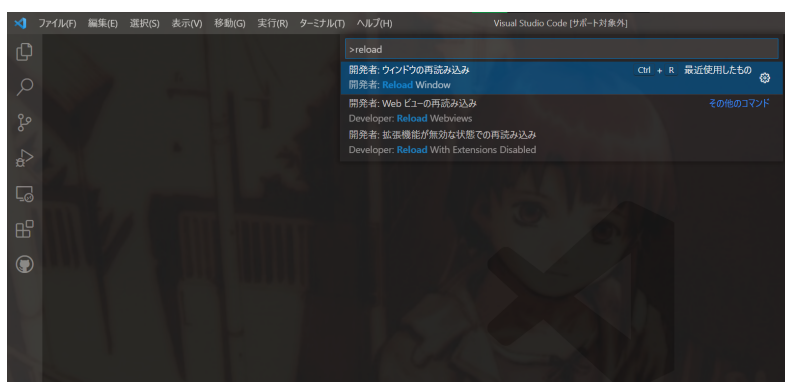


図 14

## 6.2 拡張機能について

背景画像を設定したければ、「background<sup>\*1</sup>」か「background-cover<sup>\*2</sup>」が良いです。私は「background-cover」を入れています。ただ、どちらを入れてもたまにエラーが出るようになります。特に害はなさそうですが、結構邪魔です。「Fix VSCode Checksums<sup>\*3</sup>」を入れるとそのエラーを無視することができます (果たしてそれで良いのかは微妙)。

<sup>\*1</sup> <https://marketplace.visualstudio.com/items?itemName=shalldie.background>

<sup>\*2</sup> <https://marketplace.visualstudio.com/items?itemName=manasxx.background-cover>

<sup>\*3</sup> <https://marketplace.visualstudio.com/items?itemName=lehni.vscode-fix-checksums>

## 7 おまけのおまけ

WSL 上で動かすとめっちゃくちゃ速いらしいです。いつか挑戦してみたいのですが、Linux が分からないので今の所見送っています。