

# VS Code で $\text{\LaTeX}$ を書く

## VS Code の機能を使いこなす

リュカ, 裕磨

ゼロイチゼミ, 学術サーバー

August 6, 2023

# TeXlive のインストール

- ① TeXlive のインストーラーを次のページからダウンロードする。

*https:*

*//www.tug.org/texlive/acquire-netinstall.html*

をクリックする。

- ② ページ上のリンク `install-tl-windows.exe` をクリックする。
- ③ ダウンロードを終えたら実行する。

# TeXlive のインストール

- ① 「Install」 を選択する。
- ② 「Next」 を押して進み
- ③ 「特定のミラーを選択」 では日本のミラーサイトのどれでもよいので選ぶとよい.<sup>1</sup>
- ④ インストール先は気にしなくてよい。

---

<sup>1</sup>ミラーサイトとは、TeXlive のディストリビューションを保有するサーバーを選択することである。

# 高度な設定

- 「ディレクトリ」は無変更
- 「選択したもの」欄で変更2つ行う。
  - スキーム (インストールするパッケージの種類を大まかに決める) を変更する。basic スキームを選択。
  - 下のカスタマイズに入り、言語欄から日本語と英語を追加。右の他のコレクション欄から 'LaTeX 推奨パッケージ' を追加。

設定が完了したら右下のインストールを押す。

# インストールの正常終了を確認する

インストールの後に、コマンドプロンプトを起動する。コマンドプロンプトは、Windows キーを押すと出てくるところから確認する。Mac の場合はターミナルから行う。ここに

```
latex -v
```

と打ち込めばよい。

# なぜ L<sup>A</sup>T<sub>E</sub>X を使うのか

L<sup>A</sup>T<sub>E</sub>X とは、文書執筆ツールとして使われる、組版処理系の一つである。  
L<sup>A</sup>T<sub>E</sub>X を使う利点をいくつか挙げる。

- 数式がきれいに書ける
  - Word で書こうとするととんでもない数のクリックと精密なエイム力を要求される<sup>2</sup>
- 見た目と論理構造を分離できる
- 修正、再利用が容易
- git でバージョン管理できる
  - 共同編集が可能
- 数学系、物理系では論文執筆のデファクトスタンダード
- 貧弱なスペックのパソコンでも編集作業がやりやすい
- 無料

---

<sup>2</sup>これは決して Word を貶しているわけではない。そのように Word を使うことが間違っているのである。Word も L<sup>A</sup>T<sub>E</sub>X も同じ文書執筆ツールではあるが、それぞれに長所短所があり使うべき場所というものがある。

# TeXlive Manager からパッケージのダウンロード

- ① Windows のスタートメニューから TeXliveManager を実行する。
- ② TeX live Manager を開くと現在 TeX live にあるパッケージが読み込まれる。
- ③ そして、ここから検索欄に自分の必要なパッケージを読み込むと今必要なパッケージをダウンロードすることができる。

検索欄から 'latexmk' と検索する。すると、しばし待機した後に二つの 'latexmk' が出てくる。

- latexmk
- latexmk.win32

の二つが出てくるので、これらをチェックボックスをクリックして選択項目をインストール を選択し latexmk を選択してインストールする。これで latexmk のインストールが完了する。

また今回は、LuaL<sup>A</sup>T<sub>E</sub>X を用いるので、lualatex-math というパッケージをダウンロードする。

# VSCode のインストール

Visual Studio Code は次のページからダウンロードするとよい。

*<https://code.visualstudio.com/download>*

をクリックしてそのあとの画面の中から今回の場合は、自分の環境に合わせて (Windows か Mac の) ファイルをダウンロードする。ファイルを開き、追加タスクの変更は触らないようにして Next を押し続ければインストールできる。



# VS code を日本語化する

VS Code をショートカットから開いて、積み木のようなアイコン  
Extentions（拡張機能）をクリック。検索窓に Japanese と打って、  
Japanese Language Pack for Visual Studio Code を選択してインストール  
する。これで日本語化が完了する。このように拡張機能を入れることで  
VS Code の機能をより使いやすいものにすることができる。

日本語化されない場合には、VSCode を再起動すると良い。または、コマ  
ンドパレットを出現させる。これは Ctrl+Shift+P を押すと出現する。こ  
こで、language と検索して表示言語を日本語 (Japanese) にすることでも  
日本語化することができる。

これで VSCode の日本語化は終了である。

# L<sup>A</sup>T<sub>E</sub>X を打つ

適当なファイルを生成する。そこに.tex ファイルを作成する。そこに下のコードを入れる。

Listing 1: HelloWorld

```
1  \documentclass{ltjsarticle}
2  \begin{document}
3
4  \title{はじめての\LaTeX }
5  \author{Meidai}
6  \maketitle
7  \section{はじめての\LaTeX Lua\LaTeX }
8  \subsection{小見出し！ }
9  Hello world!
10  今日は\LaTeX を覚えていってください。
11  \LaTeX + VSCode は最強の組み合わせ。
12 \end{document}
```

# L<sup>A</sup>T<sub>E</sub>X のコンパイルの仕方

L<sup>A</sup>T<sub>E</sub>X のコンパイルをコマンドラインで行う。

- ① VSCode で.tex ファイルを開く。
- ② Ctrl+@を押してコマンドラインを開く
- ③ コマンドラインに次のコマンドを打つ。

## Listing 2: latex コンパイル

---

```
1  lualatex ファイル名
```

---

これを打つと latex の中間ファイルを生成し、pdf を生成する。

# latexmk とは何か？

以上のようにコマンドを毎回打たなければならない。そこで latexmk とは、各種 L<sup>A</sup>T<sub>E</sub>X のビルドコマンドを自動で一括で実行してくれるツールである

# latexmk の設定

- ① ホームディレクトリ<sup>3</sup>直下に`.latexmk`というテキストファイルを作成する。VSCodeで開く。
- ② 次のコードを入力する。

Listing 3: latexmk の設定

```
1  # LuaLaTeX のビルドコマンド
2  $lualatex = 'lualatex %0 -synctex=1 -interaction=
    nonstopmode %S';
3  # Biber, BibTeX のビルドコマンド
4  $biber = 'biber %0 --bblencoding=utf8 -u -U --
    output_safechars %B';
5  $bibtex = 'pbibtex %0 %B';
6  # makeindex のビルドコマンド
7  $makeindex = 'upmendex %0 -o %D %S';
8
9  $pdf_mode = 4;
```

<sup>3</sup>PC>ローカルディスク>ユーザー>ユーザー名、と辿れるディレクトリのこと

# LaTeX workshop のインストール

LaTeXworkshop とは、VSCode の拡張である。VSCode 上で L<sup>A</sup>T<sub>E</sub>X を使ううえで必須の拡張である。

- ① 左の拡張機能アイコンをクリックする。
- ② 拡張機能の検索欄から `latexworkshop` と入力する。
- ③ `latexworkshop` を選び拡張機能をダウンロードする。

`latexworkshop` の設定は、`setting.json` に記述する。VS Code の設定は、`setting.json` というファイルに記載されている。

# setting.json について

setting.json を開く方法はいくつかある。

## 1 回目

VS Code 左下の設定マーク（歯車マーク）をクリックして、「設定」を選択する。右上端にあるファイルに矢印がついたアイコンをクリックする。

## 2 回目

キーボードのショートカットキーを用いて、`ctrl+shift+P` と入力することで、コマンドパレットを出現させてそこに `Preferences:Open User Setting` と打ち込む。

## 3 回目

`Ctrl+,` で設定画面を開く。さらに右上端にあるファイルに矢印がついたアイコンをクリックする。

## setting.json の記述

setting.json を開いたら、{}の中に Git Hub の配布資料に記載された LaTeX workshop の設定をコピーして書き加え、保存する。  
もし開いた時点で{}以外に何か書き込まれていた場合、{}の最後の要素にコンマをつけて、設定をコピーして書き加え、保存する。



# SyncTeX とは何か？

SyncTeX とは、TeX のソースファイルと PDF でカーソルの位置を同期する機能のこと。これを使うことで、PDF の部分が TeX のソースコードのどの位置を指し示しているかと TeX のソースコードが PDF のどの部分を構成しているのかということがわかるものである。

この資料についているコードはすでに SyncTeX に対応している。そのため、PDF の部分で Ctrl キー+クリックをすると、TeX のソースコードにジャンプする。逆に TeX の資料では、Ctrl+Alt+J を押すと PDF 資料に対応する場所が選択される。

# Ultra Math Preview のインストール

Ultra Math Preview は、LaTeX Workshop よりも強力な数式プレビューができる拡張機能である。パッケージで定義されたコマンドも Preview することができる。さらにユーザー定義の mathPreview を導入することができるために LaTeX Workshop 標準のプレビューよりもより利便性が高いものになっている。この mathPreview は Markdown でも用いることができる。

- 数式を打つとプレビューが即座に出てくる
- LaTeX と Markdown で同じマクロがプレビューに使える
- プレビューが透過できる
- プレビューの上からその下にある文字をクリックすることができる
- 拡張性が高い

拡張機能 Ultra Math Preview は、Ctrl+Shift+X で出てくる拡張機能の検索欄に Ultra Math Preveiw とうち選択してインストールを押すことでインストールできる。

# Ultra Math Preview の設定

設定方法は、配布資料に記載のソースコードを入れると設定できる。設定を参考にして L<sup>A</sup>T<sub>E</sub>X のマクロを記述することでたいいのパッケージのコマンドをプレビューすることができる。