

LATEX 環境構築勉強会

2025.04.22 D2高成

■ TeXで修論を書くメリット

① レイアウト崩れが起きにくい

- Wordの図表どっか飛んでいくストレスからの解放

② 図表番号・目次・参考文献の自動整形が強力

- Fig. ~, Table ~, 参考文献^[1]などの番号を自動で振ってくれる、目次も自動生成

③ 複雑な数式などが簡潔かつ綺麗に書ける（？）

- LaTeXは数式組版の世界標準、LaTeXの数式に慣れている人はWordの数式が汚く見える

最初のハードルはあるけれど、長文・数式・図表を含む論文には最適

■ 勉強会の目的

- 修論書き始めのタイミングで環境構築は少し大変、今のうちに
- 環境統一により、相互サポートしやすくなる

■ 今日のゴール

- PDFが自動生成・プレビューできるところまで
- ミーティング資料程度の短い文書が作成できる基盤を作る
- (必要あれば夏頃にM2向け修論テンプレ配ります)

○ TeX ディストリビューション：TeX Live

ここからインストール (<https://www.tug.org/texlive/>)

○ エディタ：Visual Studio Code + LaTeX Workshop (拡張機能)

ここからインストール (<https://code.visualstudio.com/>)

○ ビルド：latexmk + uplatex

○ 出力先：out/ディレクトリに出力する設定にしています

※ ローカル環境をあまり汚したくない人は、Dockerと呼ばれるコンテナ型の仮想環境を使う方法もあります。PCや開発環境にある程度慣れている人は挑戦してみても良いかもですが、本構成に比べて若干使い勝手は劣ると思います。

本資料作成の参考（閲覧：20250419）

- 【大学生向け】LaTeX完全導入ガイド Windows編 (2022年)
- VSCODE で最高の LaTeX 環境を作る

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 .latexmkrc と setting.json の設定

STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 .latexmkrc と setting.jsonの設定

STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

STEP 0 | 旧ver. のTeXの確認

6

⚠ TeX Live/MacTeX は毎年リリースされます。複数のTeX Liveがインストールされるとバージョンの競合が起こりエラーが出ることがあるため、旧ver. の有無を確認します。

Windows

1. コマンドプロンプトを開く
(Winキー + R に cmd を入力)

2. 右のコマンドを入力

```
where uplatex  
where pdflatex  
where xelatex  
where lualatex
```

3. 結果例

インストール済：

```
> where uplatex  
C:\texlive\2024\bin\win32\uplatex.exe
```

未インストール：

```
> where uplatex  
INFO: Could not find files for the given pattern(s).
```

Mac

1. ターミナルを開く
(⌘ + space に terminal を入力)

2. 右のコマンドを入力

```
which uplatex  
which pdflatex  
which xelatex  
which lualatex
```

3. 結果例

インストール済：

```
$ which uplatex  
/Library/TeX/texbin/uplatex
```

未インストール：

```
$ which uplatex  
何も返ってこない or zsh: command not found: uplatex
```

旧ver. が未インストールだった人は STEP 1へ

STEP 0 | 旧ver. があった人向け削除手順

7

Windows

1. 旧ver. のフォルダを開く

例 : C:\texlive\2023\

2. フォルダ内にある以下のファイルを実行

「uninstall-tl-windows.bat」

コマンドプロンプトが開き、自動で削除処理

3. 削除完了後、フォルダが残っていれば

手動で削除

例 : C:\texlive\2023\

Mac

1. ターミナルでtexliveディレクトリへ移動

```
cd /usr/local/texlive/
```

2. 中身を確認 : ls

出力例 : 2022 2023 texmf-local

3. 以下のコマンドを実行

```
sudo rm -rf /usr/local/texlive/2023
```

- ・ 2023の部分は削除したいver.に応じて
- ・ 管理者のパスワードが求められる

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 .latexmkrc と setting.jsonの設定

STEP 6 ビルドテスト②

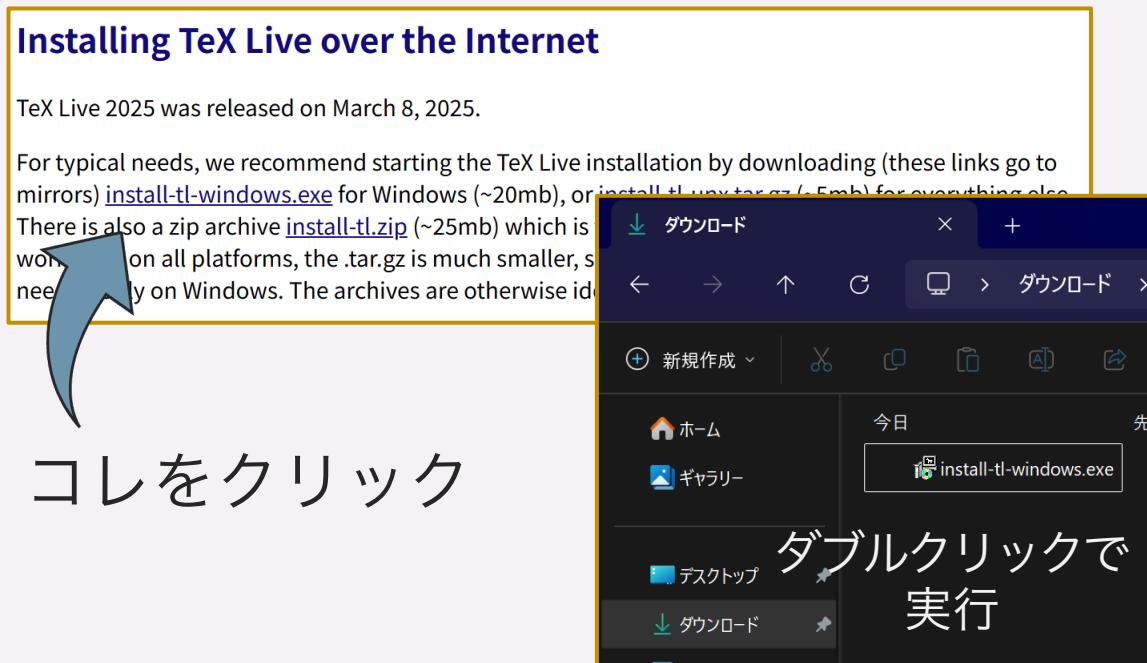
STEP 7 トラブルシューティング

STEP 1 (Win) | TeX ディストリビューションのインストール

■ TeX ディストリビューションとは

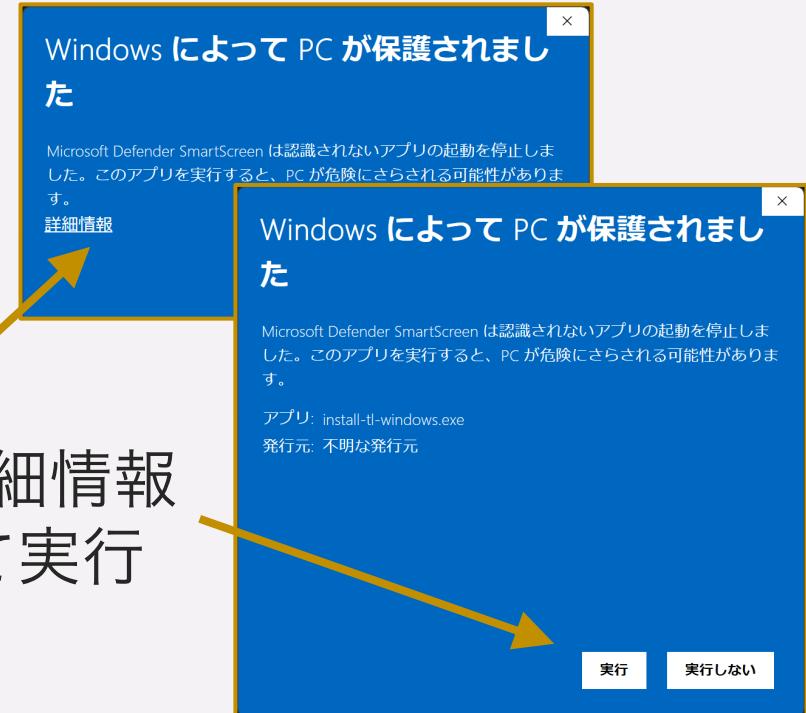
「LaTeX本体 + エンジン + パッケージ + 補助ツール類」をまとめたセット

1. TeX Users Groupのサイトにアクセスし、インストーラをDL、[実行]



コレをクリック

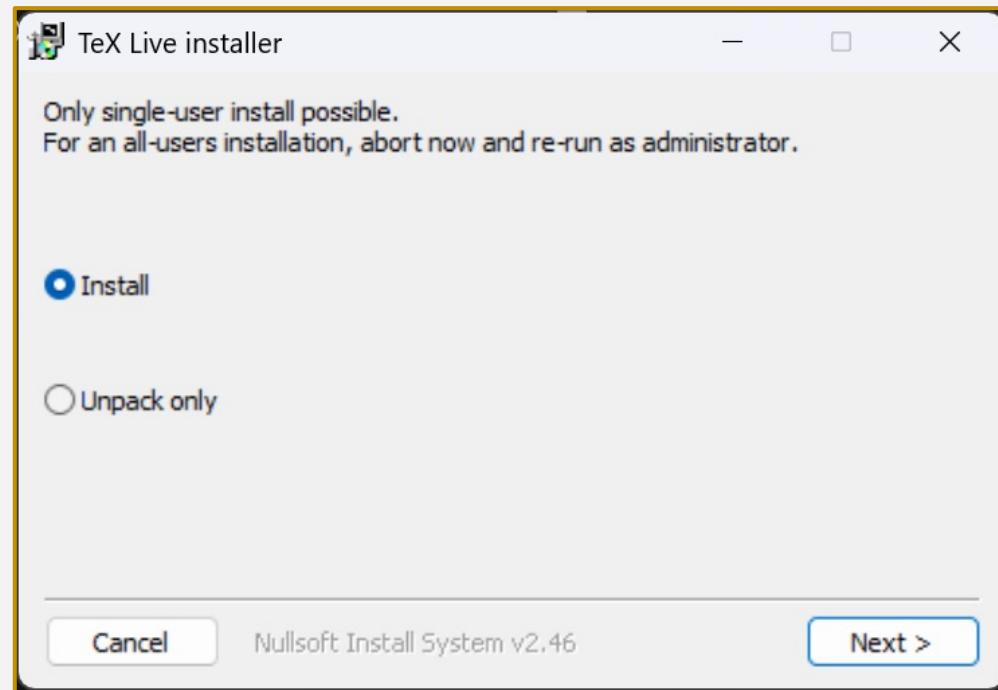
警告出たら詳細情報
クリックして実行



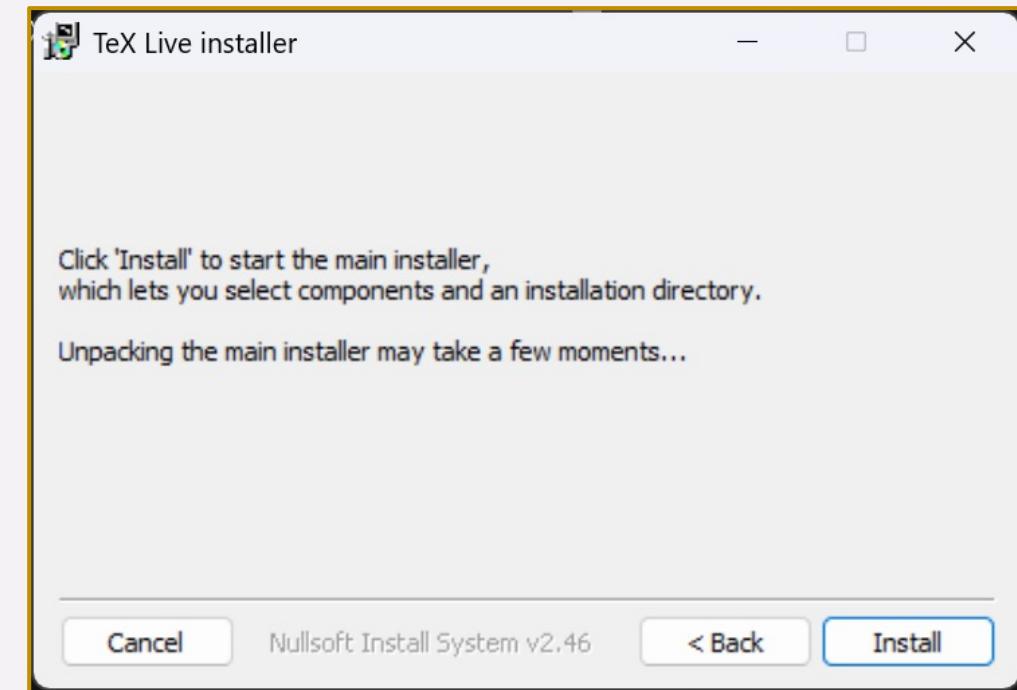
実行

実行しない

2. インストーラのインストール



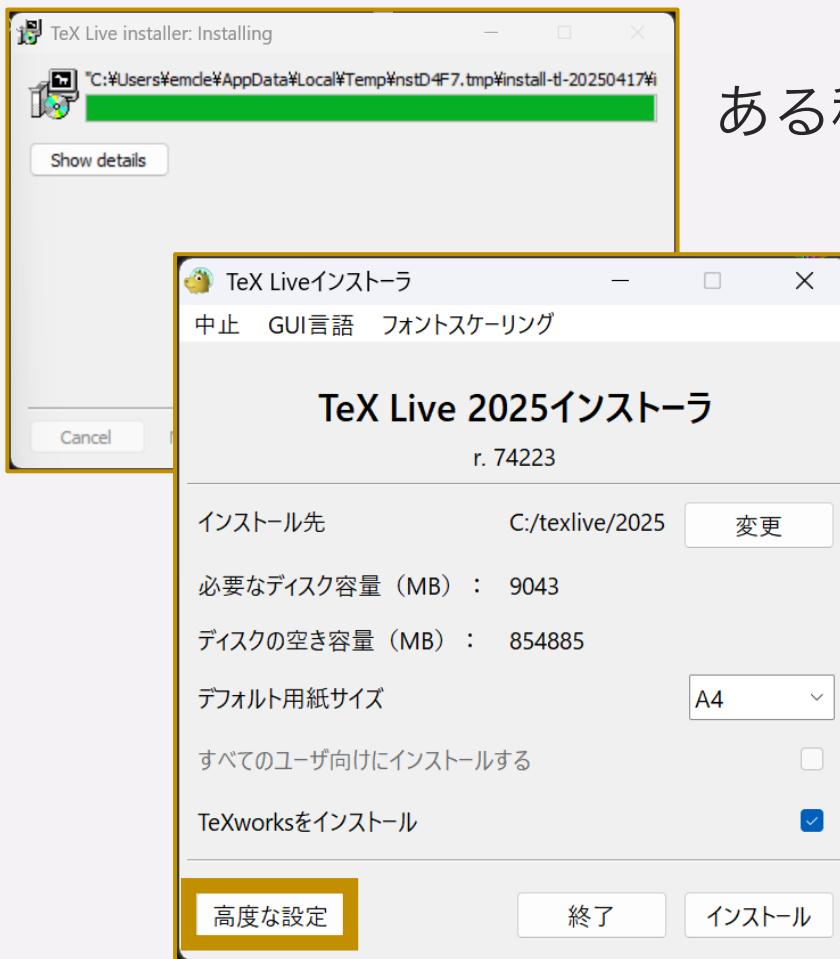
Installを選択し、[Next]



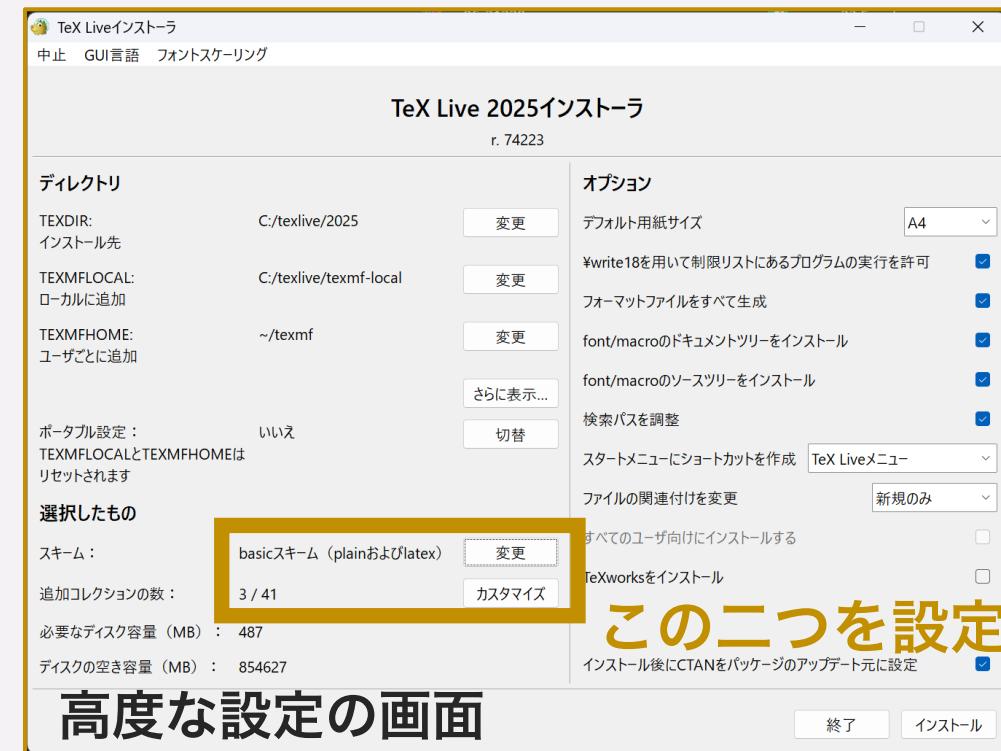
[Install]

3. 最小構成でインストールを進めます。

(フルパッケージでも良いが、容量が大きく、完了までに数時間かかる)



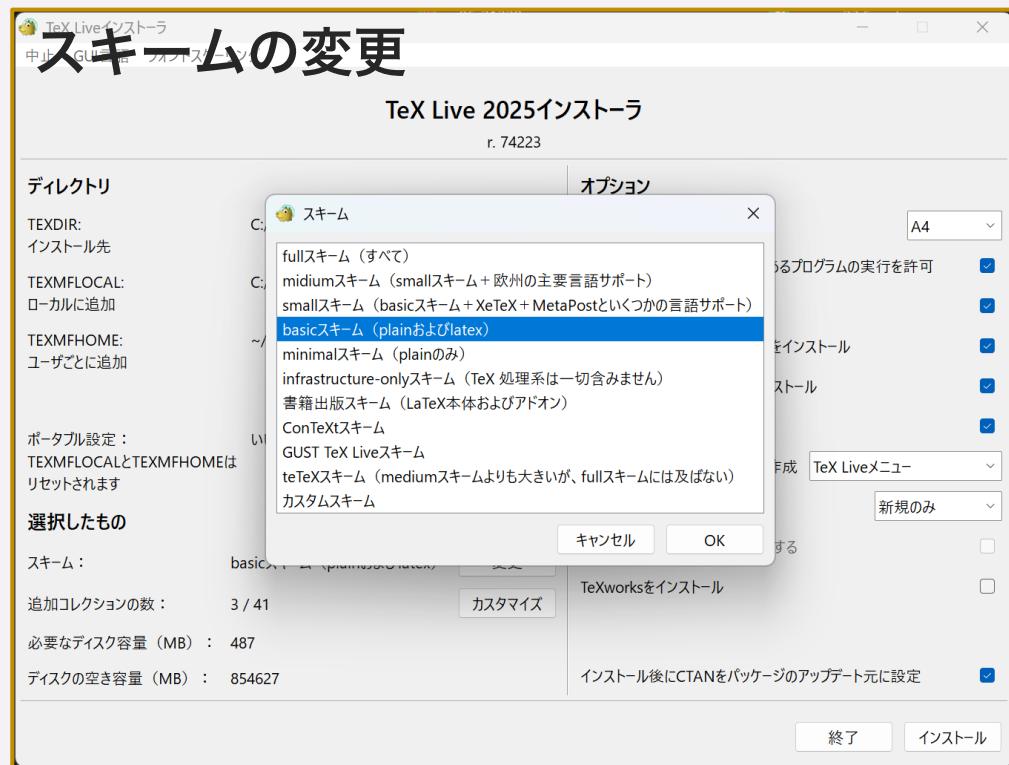
ある程度進むと別のウィンドウが開くので、**高度な設定**をクリック



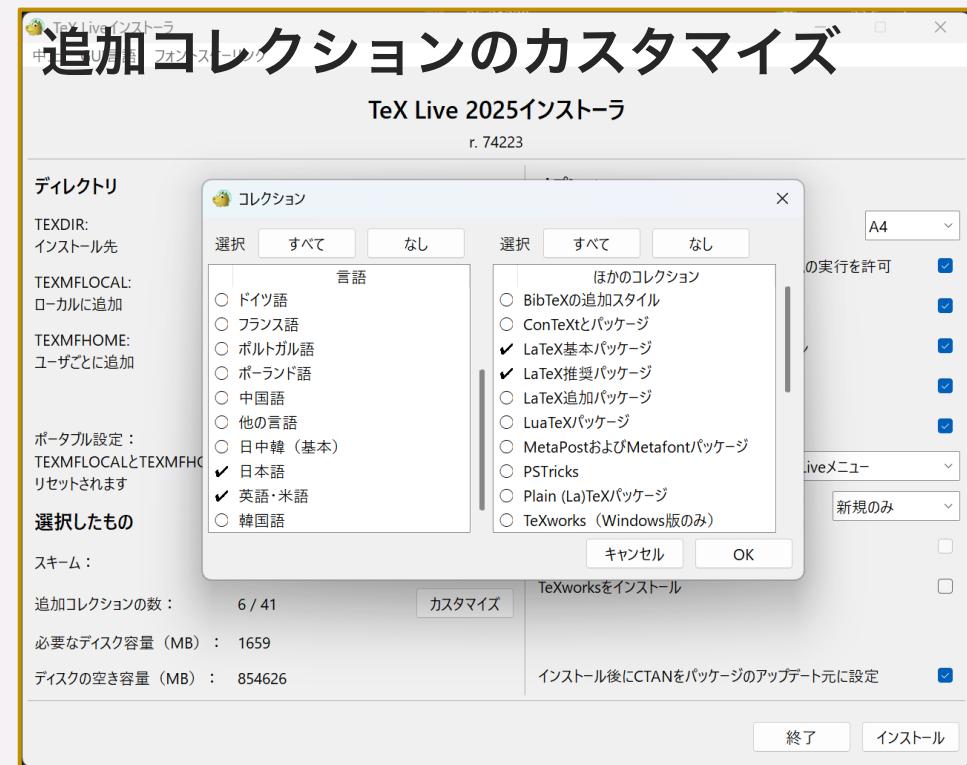
高度な設定の画面

この二つを設定します

4. スキームの変更と追加コレクションのカスタマイズ



basicスキームを選択し[OK]



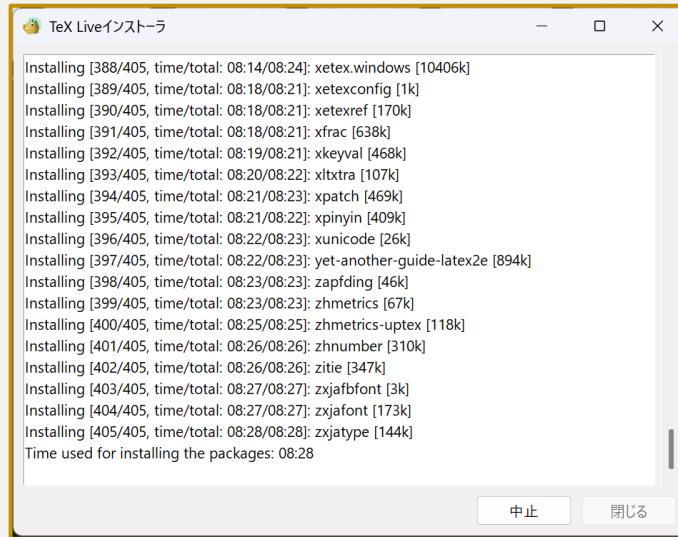
上と同じものを選択し[OK]

▶ スキーム : **basicスキーム**、追加コレクションの : **6/41** となっていればインストール

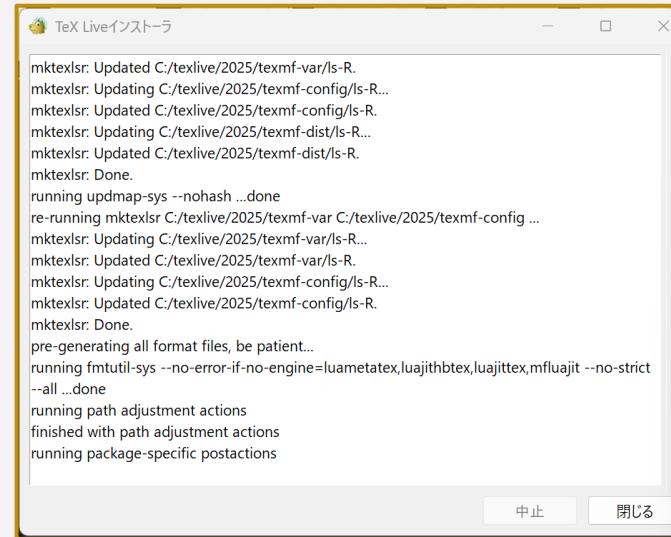
STEP 1 (Win) | TeX Live Installer : 重要箇所

13

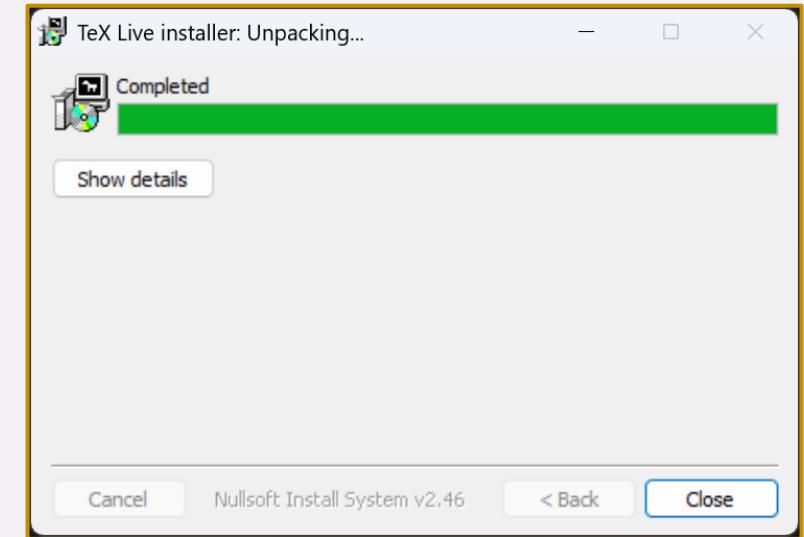
5. インストールが終わるのを待ちます。僕のPCでは10分程かかりました。



色々インストールされていく



[閉じる]が選択できるように
なれば完了

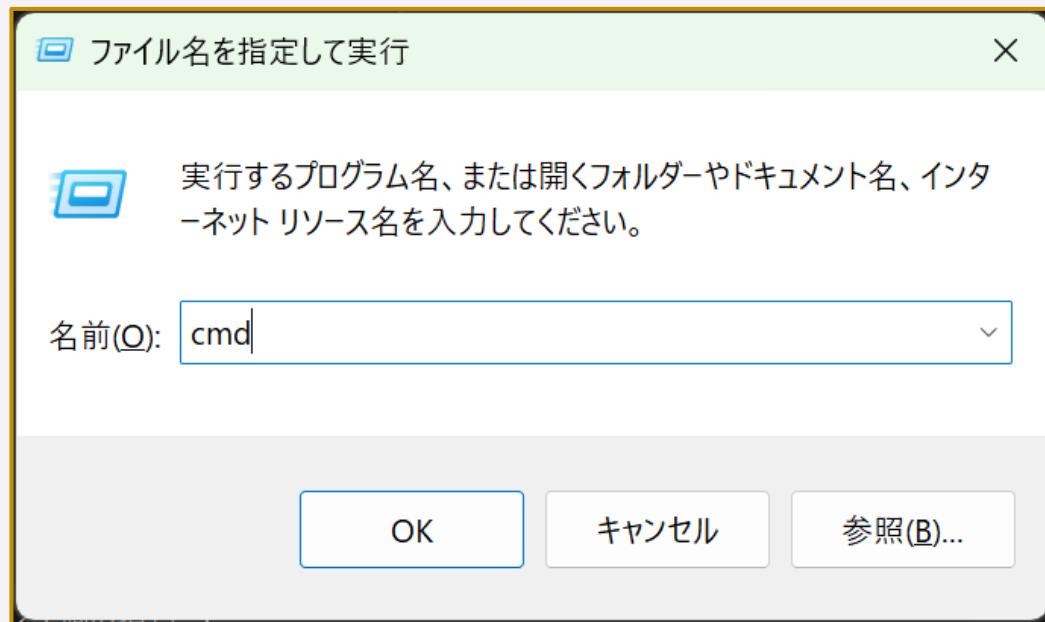


これでLaTeXのインストールは
完了です [Close]

STEP 1 (Win) | インストールできているかの確認

14

コマンドプロンプトで `latex` を入力、バージョン情報を確認する



[Win + R] で開くダイアログボックスに
cmdを入力

A screenshot of a Windows Command Prompt window titled 'cmd.exe'. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\emcle>latex
This is pdfTeX, Version 3.141592653-2.6-1.40.27 (TeX Live 2025) (preloaded format=latex)
 restricted \write18 enabled.
**
```

`latex` を入力しenter
バージョン情報が返って来ればインストール成功

where uplatax
where pdflatex
where xelatex
where lualatex

← これを試してみるのもアリ

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 以降、Windowsのスクリーンショットで

STEP 6 説明しますが、操作はMacでも一緒です。

STEP 7 PDF自動プレビュー 唯心&ファンブルフェ ノイズ

▶ 既にVS Codeを使っている人はSTEP 3へ

■ VS Code (Visual Studio Code) とは

Microsoftが開発している、無料で使える高機能コードエディタ

VS Codeを使わなくともLaTeXの文書作成は可能ですが、**補完機能やPDFプレビューなど**利便性が高いため、今回はVS Codeを利用した環境を構築します。
(Pythonなどを記述する際にも使用できるため、今後の研究にも役立ちます)

インストールはめちゃめちゃ簡単なので説明省きます。

VS Codeの[公式ページ](#)からインストールしてください。

一応参考：[Windows 11へのVisual Studio Code のインストール](#)

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 .latexmkrc と setting.jsonの設定

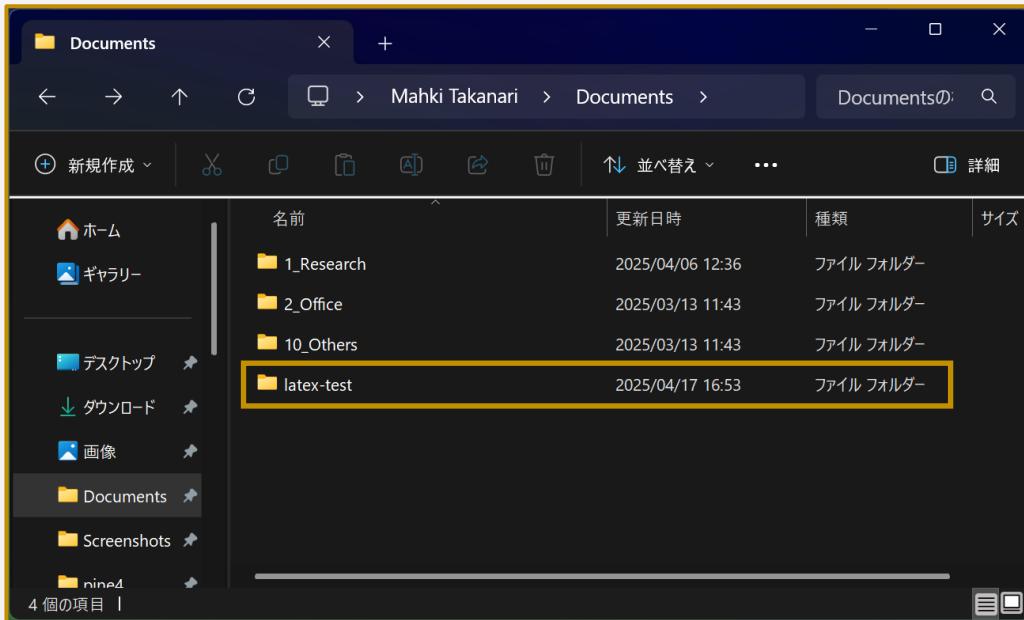
STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

STEP 3 | ビルドテスト①

18

1. ディレクトリの作成



任意の場所に、任意の名前の
新規ディレクトリを作成してください

■ フォルダ名・ファイル名の付け方について
研究室のグループLINEで以前伝えましたが、

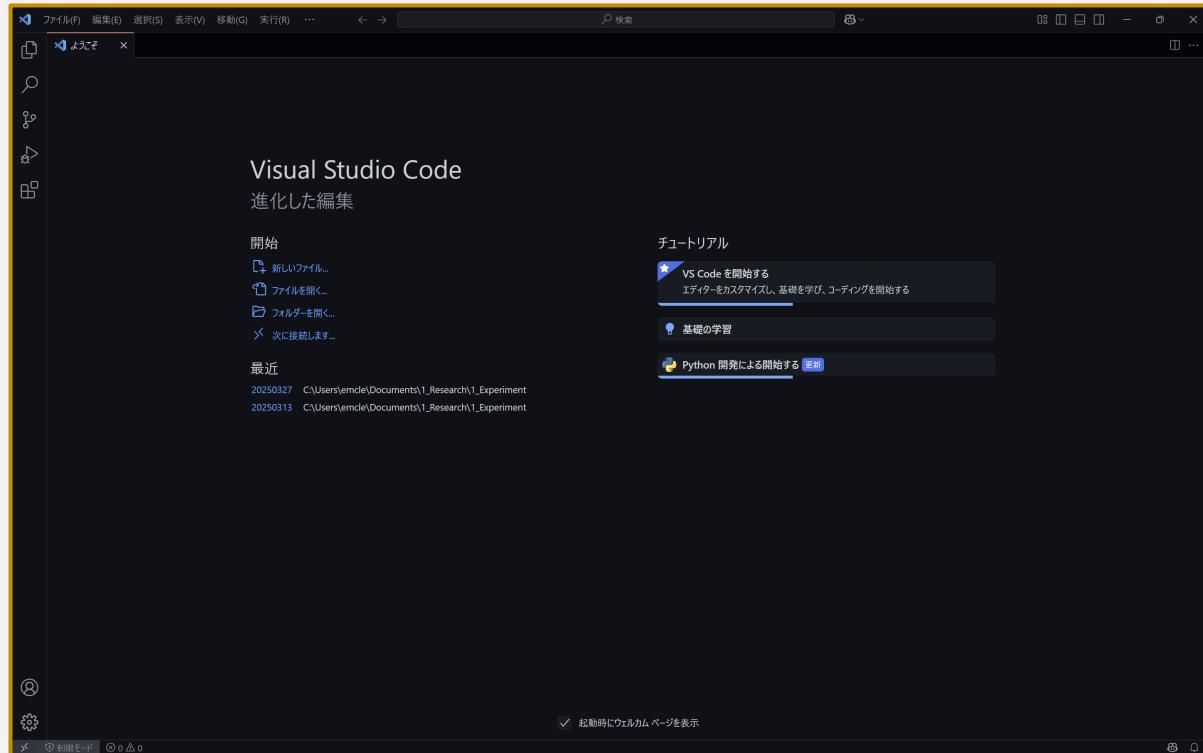
- 英数字(半角)のみを使用する
 - スペースは使わずに[_] or [-] を使う
- ことを推奨します。日本語文字を使用していると
- OS間・クラウドサービス間での互換性
 - コマンドラインやプログラムの処理
- で躊躇ことがあります。

LaTeXでの文書作成では特に、作業ディレクトリより上の階層に日本語（全角文字）が
含まれていると、コンパイルエラーの原因になることがあります。普段から気をつけましょう。

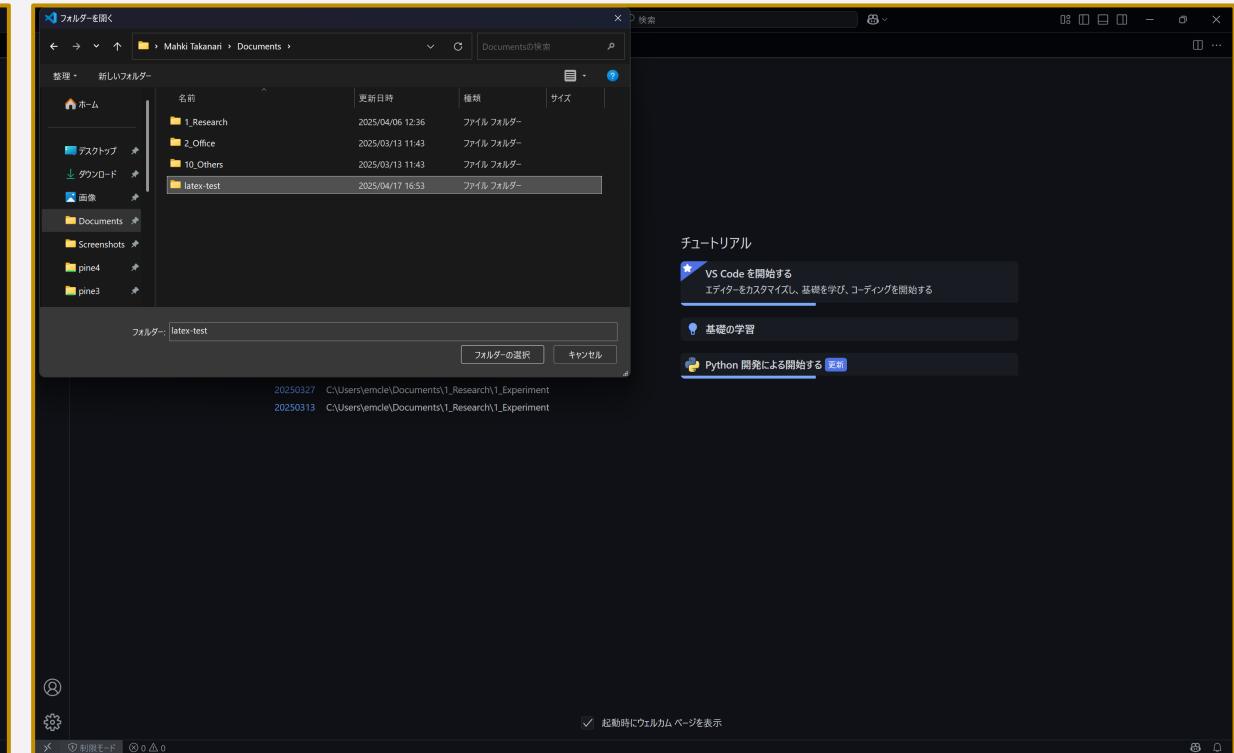
STEP 3 | ビルドテスト①

19

2. VS Codeで作業ディレクトリを開く

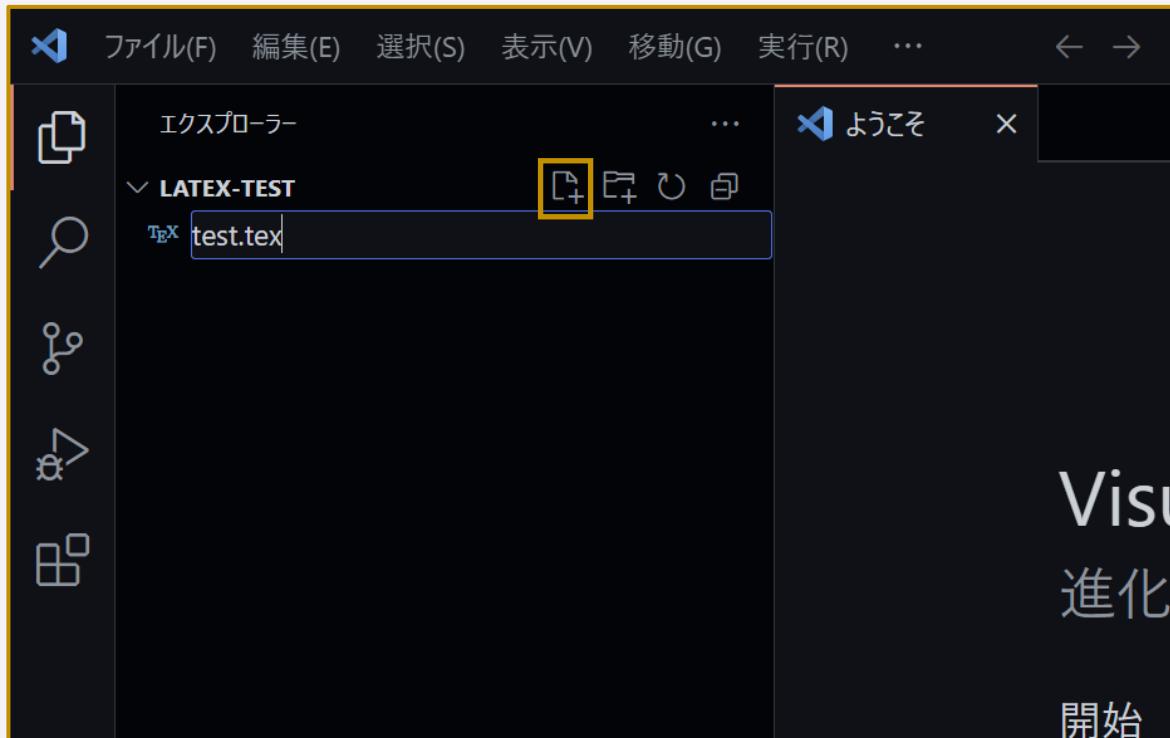


[Ctrl + O]
or
[ファイル]→[フォルダを開く]



作成したディレクトリを選択し
[フォルダの選択]

3. test.texファイルを作成



開いたディレクトリ名の右にある をクリック
「test.tex」と入力し Enter

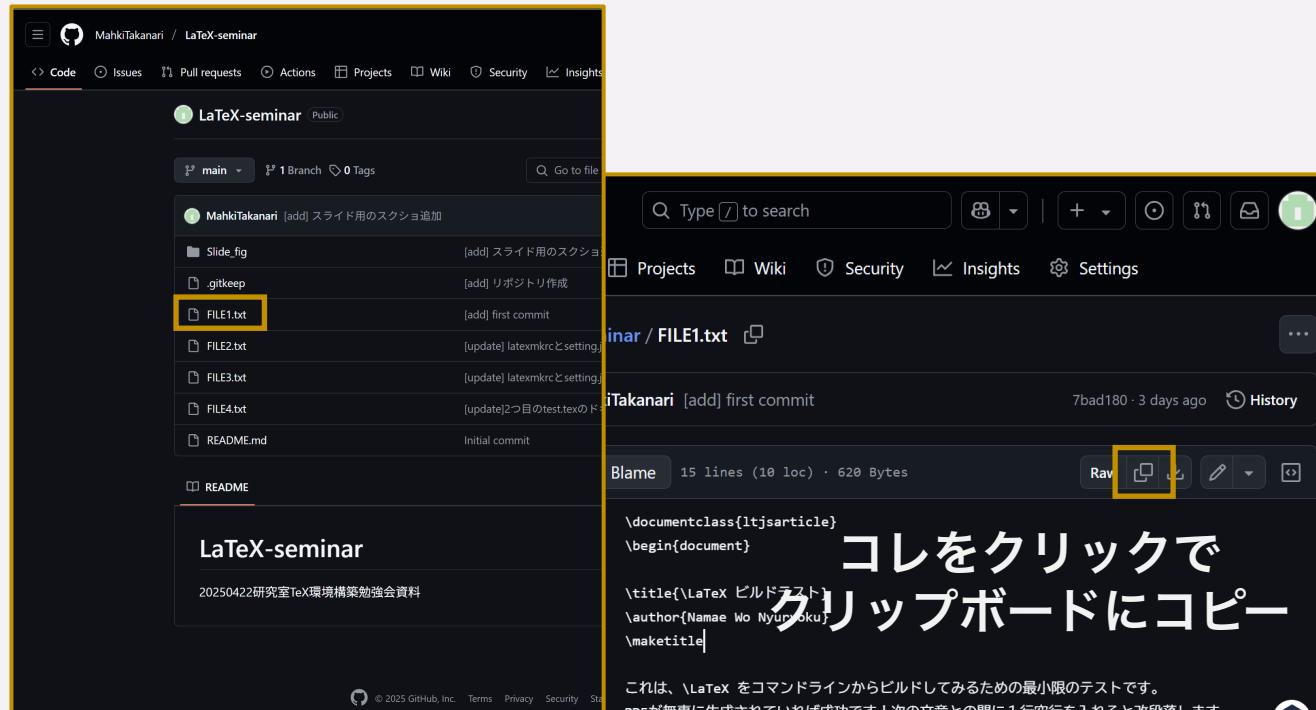
→ 新しい.texファイルが作業ディレクトリ直下に
作成され、編集画面が開きます

STEP 3 | ビルドテスト①

21

4. サンプルスクリプトを入力

GitHubのページにアクセス



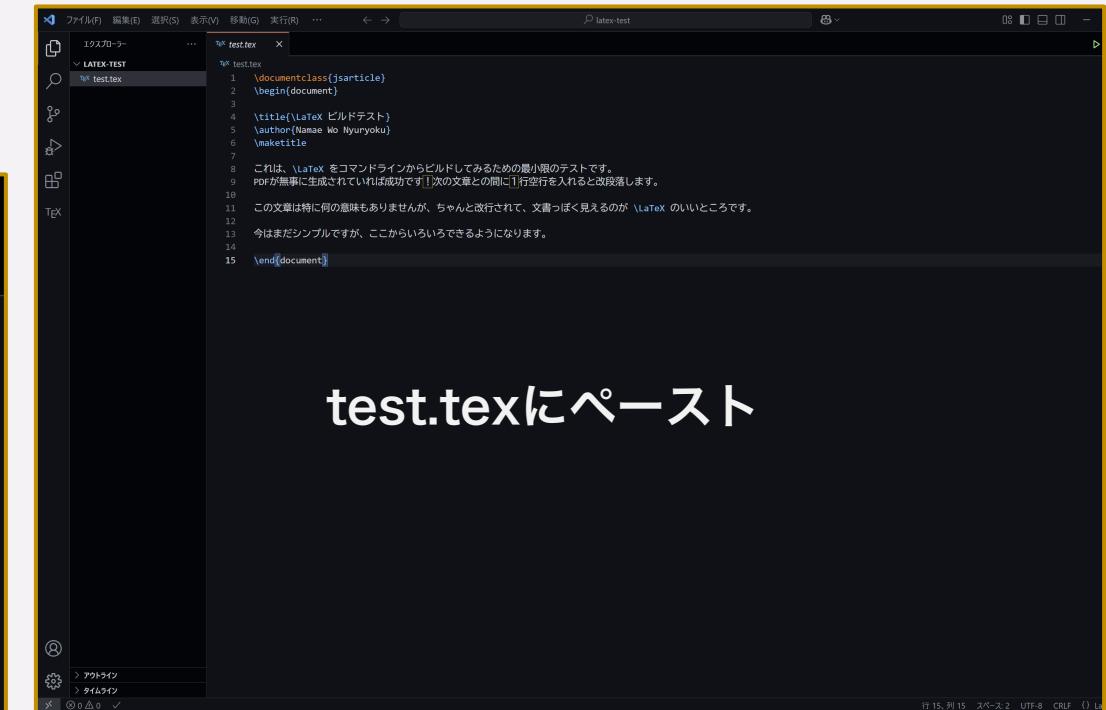
The screenshot shows a GitHub repository named 'LaTeX-seminar' belonging to user 'MahkiTakanari'. The 'FILE1.txt' file is highlighted with a yellow box. The file content is displayed below:

```
\documentclass{ltjsarticle}
\begin{document}
\title{\LaTeX ビルドテスト}
\author{Namee Wo Nyuryoku}
\maketitle
```

Below the code, there is Japanese text: "コレをクリックでクリップボードにコピー".

FILE1.txtをクリック

リンク飛べなかった人用：<https://github.com/MahkiTakanari/LaTeX-seminar>



The screenshot shows a LaTeX editor with the 'test.tex' file open. The copied code from FILE1.txt has been pasted into the editor. A yellow box highlights the 'Raw' button in the GitHub interface, indicating where the code was copied from.

```
\documentclass{ltjsarticle}
\begin{document}
\title{\LaTeX ビルドテスト}
\author{Namee Wo Nyuryoku}
\maketitle
```

Below the code, there is explanatory Japanese text: "これは、\LaTeX をコマンドラインからビルドしてみるための最小限のテストです。PDFが無事に生成されていれば成功です！次の文章との間に1行空行を入れると改段落します。この文書は特に何の意味もありませんが、ちゃんと改行されて、文書っぽく見えるのが \LaTeX のいいところです。今はまだシンプルですが、ここからいろいろできるようになります。".

test.texにペースト

行 15 列 15 スペース 2 CRLF () La

STEP 3 | ビルドテスト①

22

5. ビルド

■ LaTeXの処理ステップ

```
uplatex main.tex      # LaTeX 本文 → DVIファイル (main.dvi) を生成  
uplatex main.tex      # 参照 (\ref など) を解決するためにもう一度  
dvipdfmx main.dvi    # DVI → PDF に変換
```

LaTeXは一度のコンパイルではすべての情報が揃わないため複数回コンパイルして参照やレイアウトを正しく反映する必要があります。

補足：修論のような参考文献や索引がある場合

```
uplatex main.tex  
upbibtex main        # 文献処理 (.aux → .bb1 を生成)  
uplatex main.tex  
uplatex main.tex     # cross-reference 更新  
upmendex main        # 索引処理 (.idx → .ind)  
uplatex main.tex  
dvipdfmx main.dvi
```

The screenshot shows a LaTeX editor interface with a dark theme. On the left, there's a file tree labeled 'LATEX-TEST' containing 'test.tex'. The main area displays the LaTeX code for 'test.tex':

```
\documentclass{jsarticle}  
\begin{document}  
\title{\LaTeX ビルドテスト}  
\author{Name Wo Nyuryoku}  
\maketitle  
  
これは、\LaTeX をコマンドラインからビルドしてみるための最小限のテストです。  
PDFが無事に生成されれば成功です！次の文書との間に1行空行を入れると改段落します。  
この文章は特に何の意味もありませんが、ちゃんと改行されて、文書っぽく見えるのが \LaTeX のいいところです。  
今はまだシンプルですが、ここからいろいろできるようになります。  
\end{document}
```

Below the editor is a terminal window titled 'powershell' showing the command:

```
(base) PS C:\Users\emcle\Documents\latex-test> uplatex test.tex
```

ターミナルの開き方

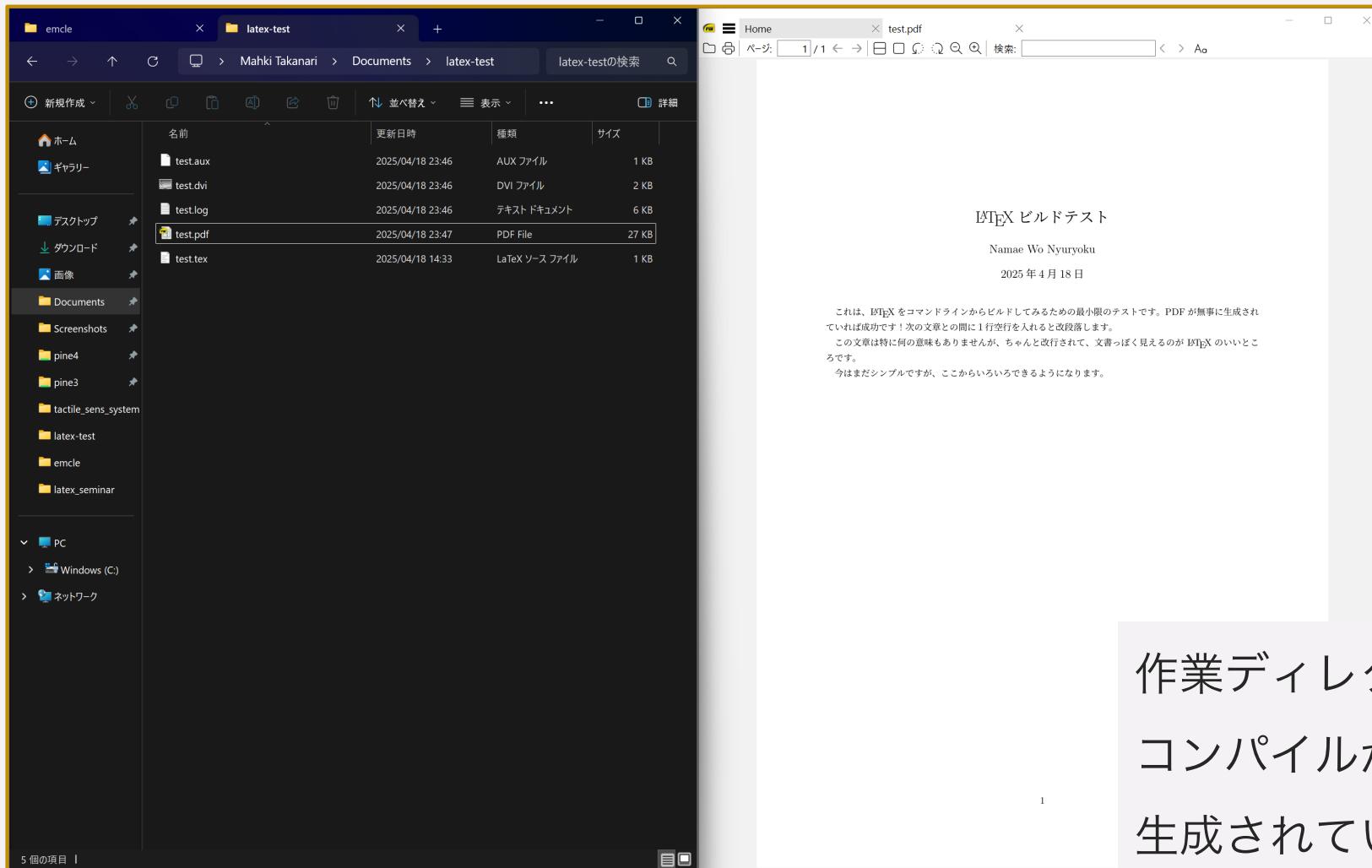
Win : [Ctrl + `] (バッククォート)
Mac : [⌘ + `]

ターミナルからコンパイル
uplatex test.tex
uplatex test.tex
dvipdfmx test.dvi を順に実行 (タイプに注意！)

STEP 3 | ビルドテスト①

23

6. 出力されたPDFを確認



作業ディレクトリを確認してみてください。
コンパイルが上手くいっていれば test.pdf が
生成されているはずです！！

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 .latexmkrc と setting.jsonの設定

STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

STEP 3で紹介したビルド方法は最も基本的な手順ですが、ビルド支援ツール [latexmk](#) と拡張機能 [LaTeX Workshop](#) を導入することで、より快適に作業できるようになります。

■ latexmk

LaTeX文書のビルドを自動で最適な回数だけ実行してくれるツール

```
uplatex main.tex  
upbibtex main  
uplatex main.tex  
uplatex main.tex  
dvipdfmx main.dvi
```

これが

```
latexmk -e “$latex=q/uplatex -kanji=utf8 -no-guess-input-enc/” -pdfdvi main.tex
```

の一発で済むようになる

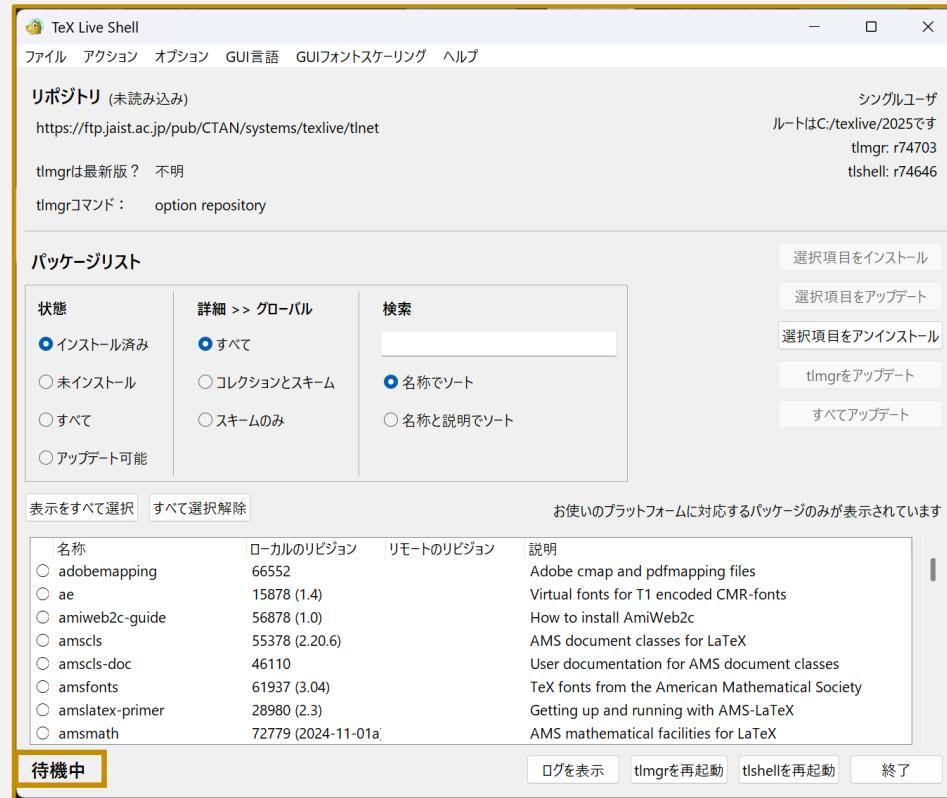
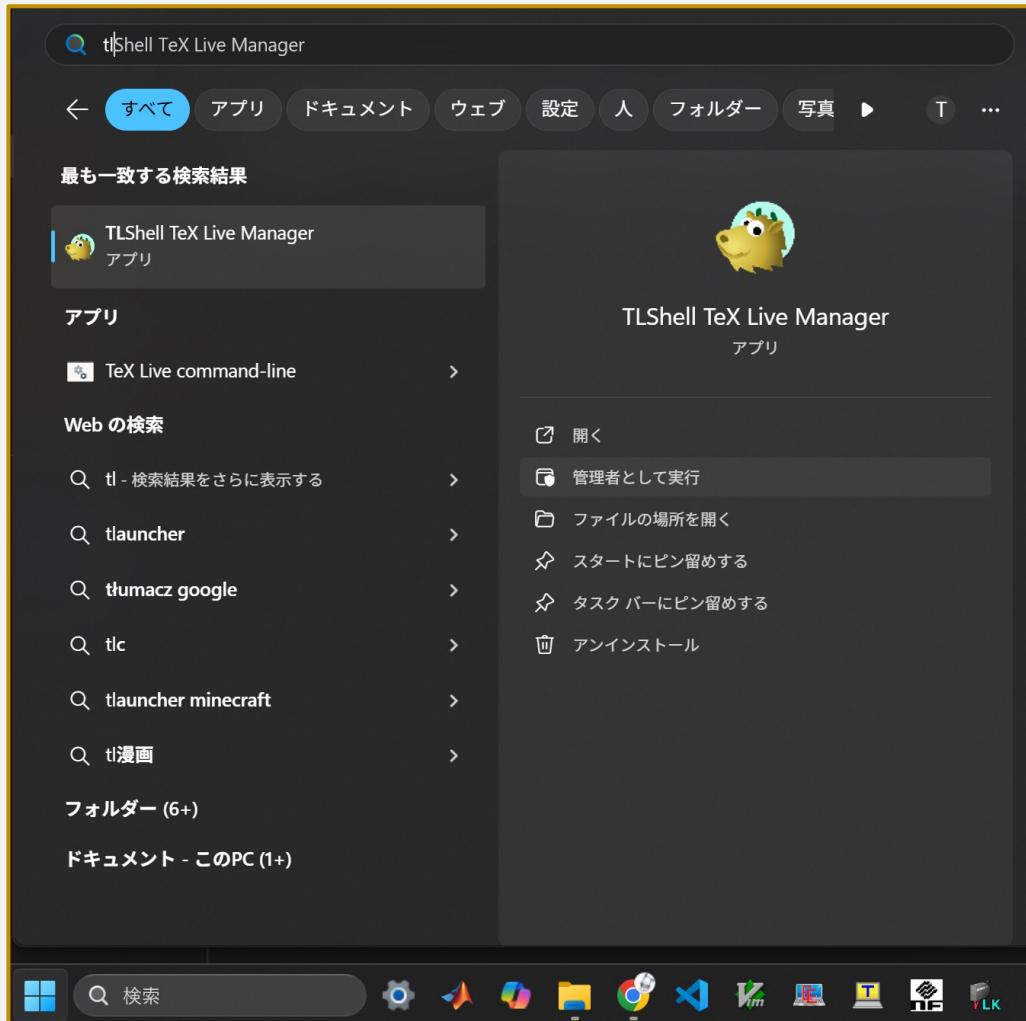
■ LaTeX Workshop

VS Codeの拡張機能である総合支援ツール

- 補完
- ビルド
- プレビュー
- エラー表示

が、すべてVS Code内で完結

1. スタートメニューからTLShell TeX Live Managerを実行



起動後は左下が待機中になるまで待つ

STEP 4 (Win) | latexmkの導入

27

2. latexmkのパッケージをインストール



[ファイル]

→ [リポジトリを読み込む]



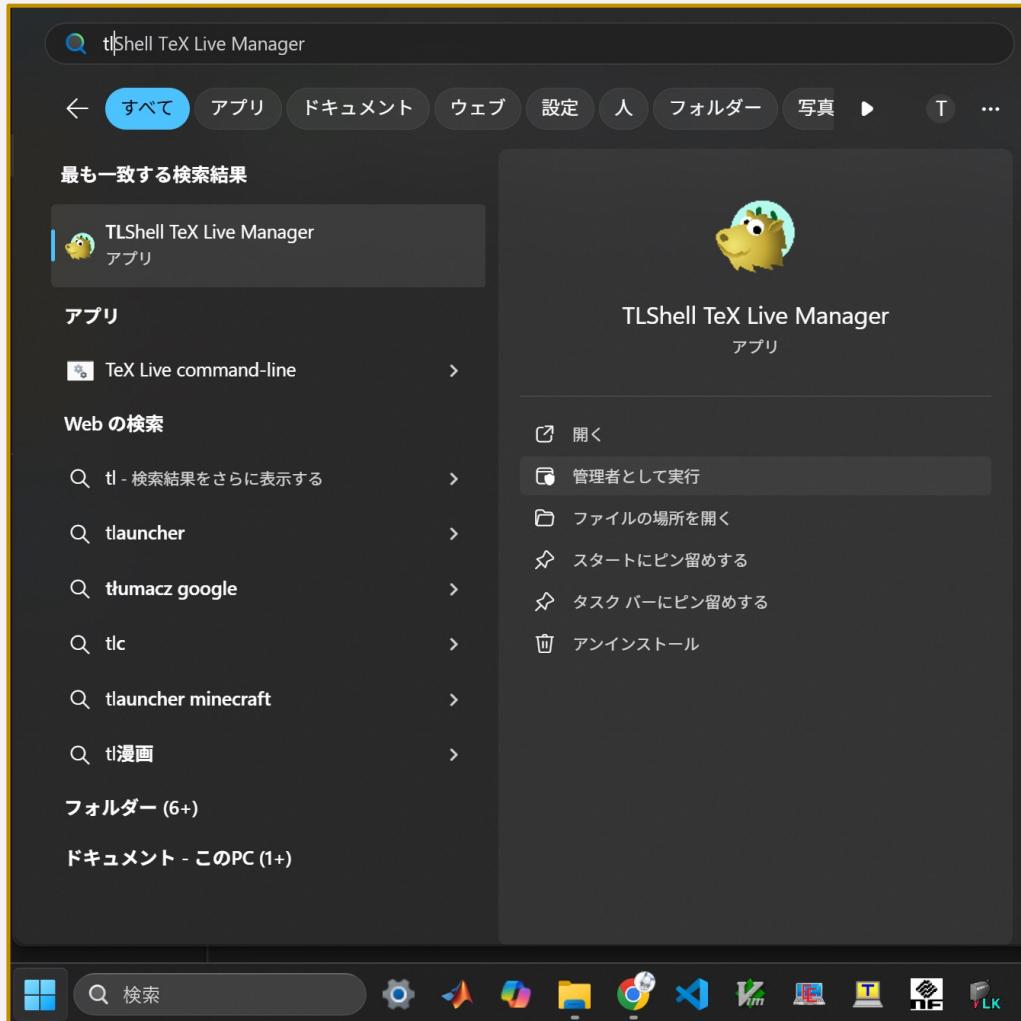
パッケージリストの状態を
「すべて」にして
検索窓に

「**latexmk**」を入力

- **latexmk**
 - **latexmk.windows**
- にチェックを入れて
[選択項目をインストール]

STEP 4 (Win) | latexmkの導入

28

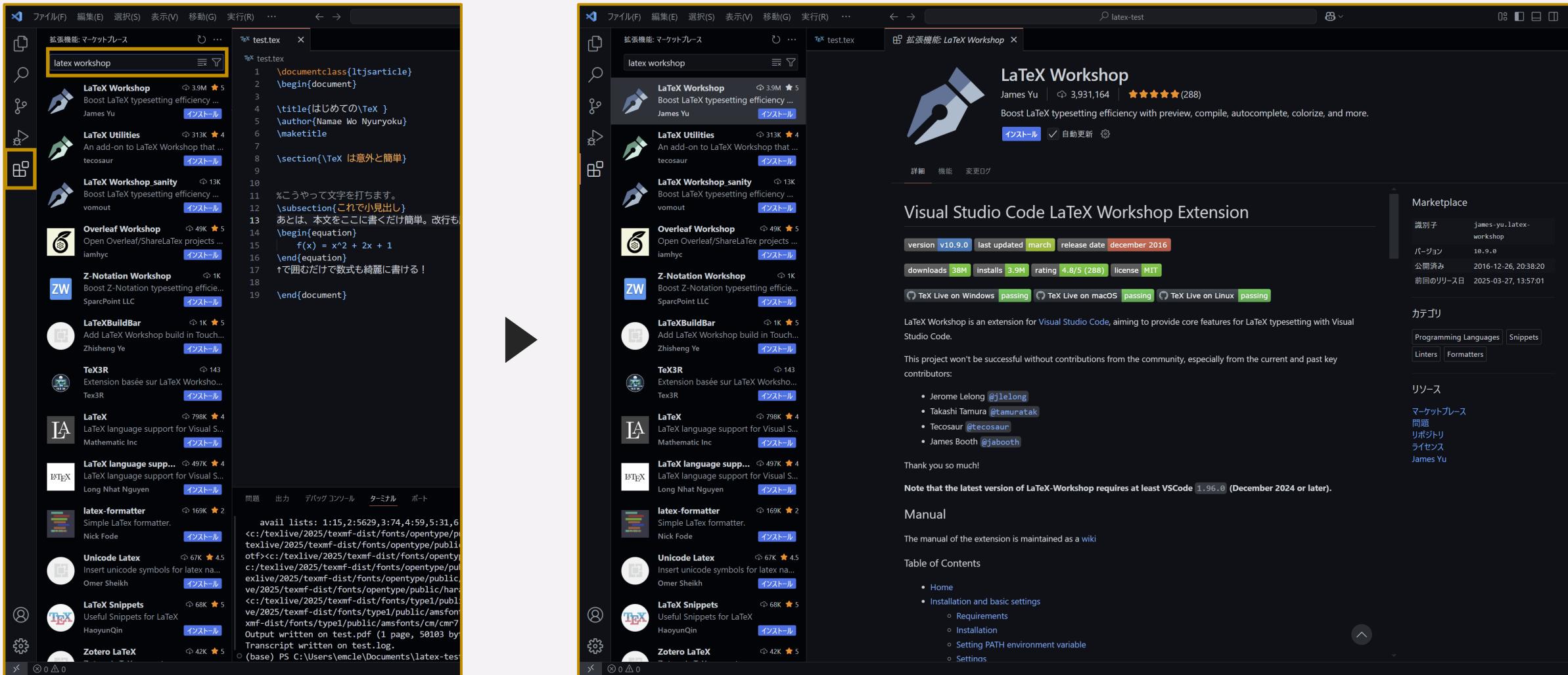


このように、TLShell TeX Live Manager は
パッケージを追加・管理する際に使用します。
今後、パッケージのインストールが必要になった場合は
同様の手順を踏んでインストールを行なってください。

ちなみに、コマンドプロンプトから
`tlmgr install [パッケージ名]`
を実行することでも、パッケージの追加が可能です。

STEP 4 | LaTeX Workshopの導入

29



左メニューにある積み木アイコンをクリック → 検索窓[latex workshop] → インストール

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 ビルド支援ツールの導入

STEP 5 **.latexmkrc と setting.jsonの設定**

STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

STEP 5 | .latexmkrc と setting.json の設定

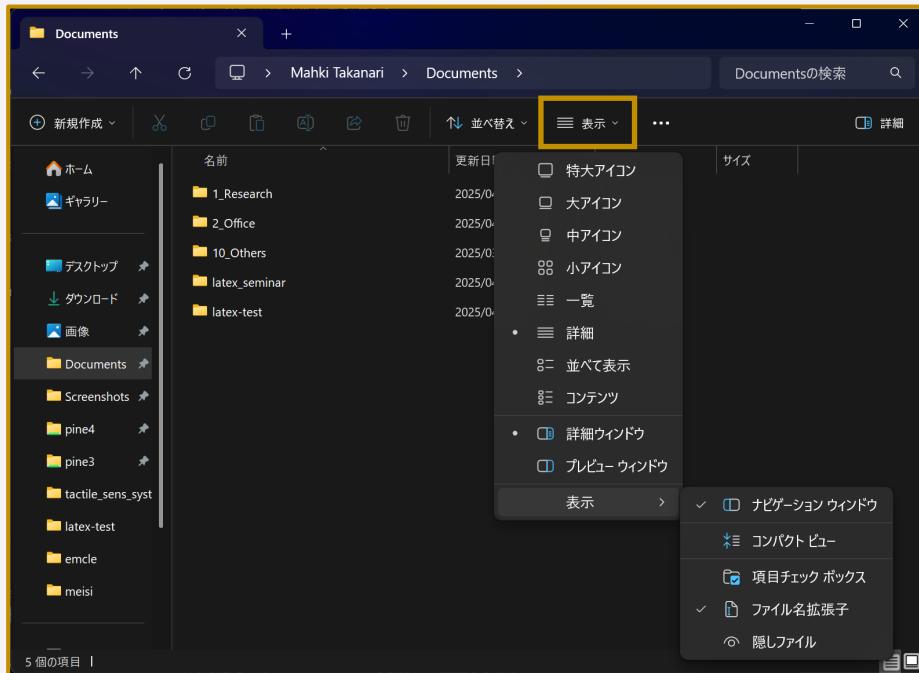
31

ファイル名拡張子を表示する設定 - これをしておくと安全に次の工程が行えます

Windows

エクスプローラーを開いて

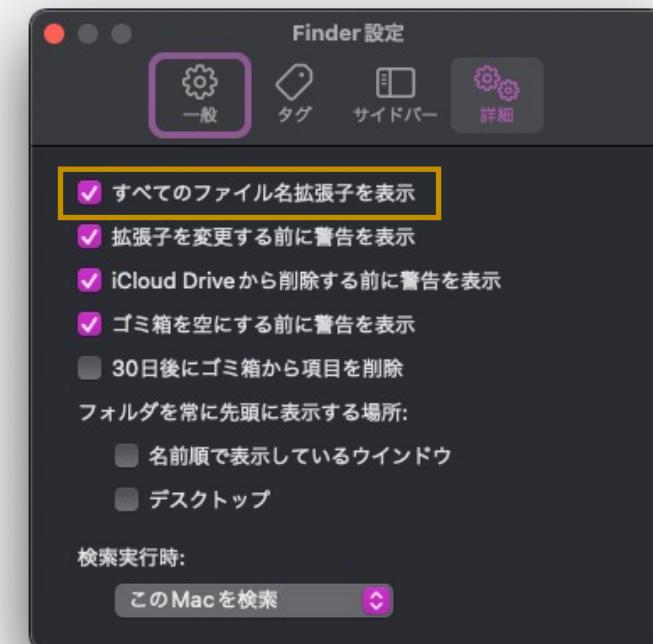
表示 → ファイル名拡張子



Mac

Finder を開いて [⌘ + ,]

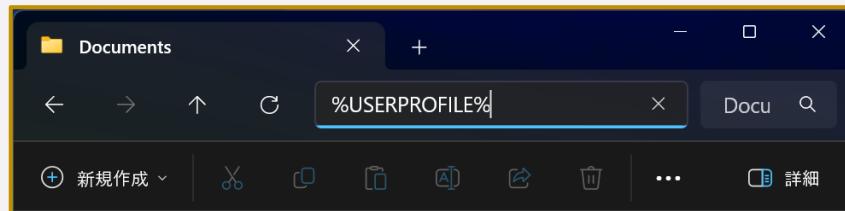
→ すべてのファイル名拡張子を表示



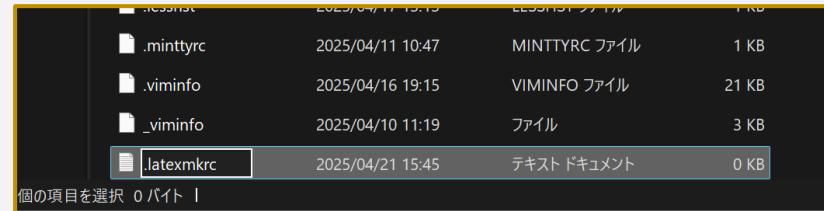
■ .latexmkrc とは

latexmkの動作をカスタマイズするための設定ファイル。 latexmkコマンド実行時に自動で読み込まれる。置き場所に任意性なし（仕組みを理解してれば変えても良いです）、ホームディレクトリ直下に配置します。

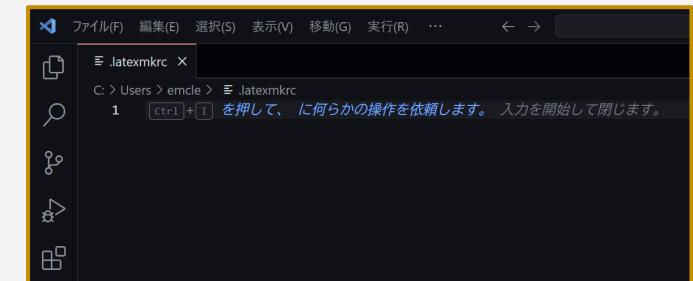
1. .latexmkrcファイルを作成し、開く



エクスプローラーのアドレスバーに
%USERPROFILE%
と入力しEnter
→ ホームディレクトリに移動



新規テキストドキュメントを作成。
ファイル名を
.latexmkrc
とします。 タイポ注意！

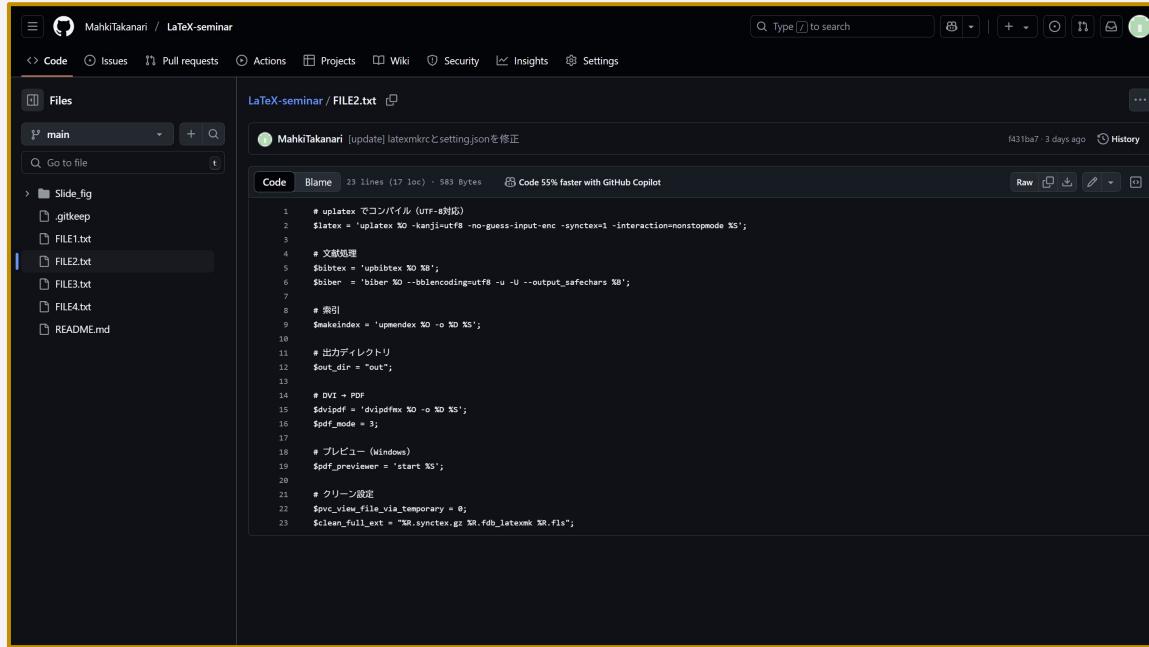


.latexmkrcファイルを開きます。
画像はVS Codeを使っていますが
任意のテキストエディタで構いません

STEP 5 | .latexmkrcの設定

33

2. .latexmkrcを記述する



The screenshot shows a GitHub repository interface for 'MahkiTakanari / LaTeX-seminar'. The 'FILE2.txt' file is open, displaying the .latexmkrc configuration script. The code is as follows:

```
# uplatex でコンパイル (UTF-8対応)
$uplatex = 'uplatex %O -kanji=utf8 -no-guess-input-enc -synctex=1 -interaction=nonstopmode %S';

# 文献処理
$bibtex = 'upbibtex %O %B';
$biber = 'biber %O --bbencodingsutf8 -u -U --output_safechars %B';

# 索引
$makeindex = 'upmendex %O -o %D %S';

# 出力ディレクトリ
$out_dir = "out";

# DVI → PDF
$dvipdfm = 'dvipdfmx %O -o %D %S';
$pdffile_mode = 3;

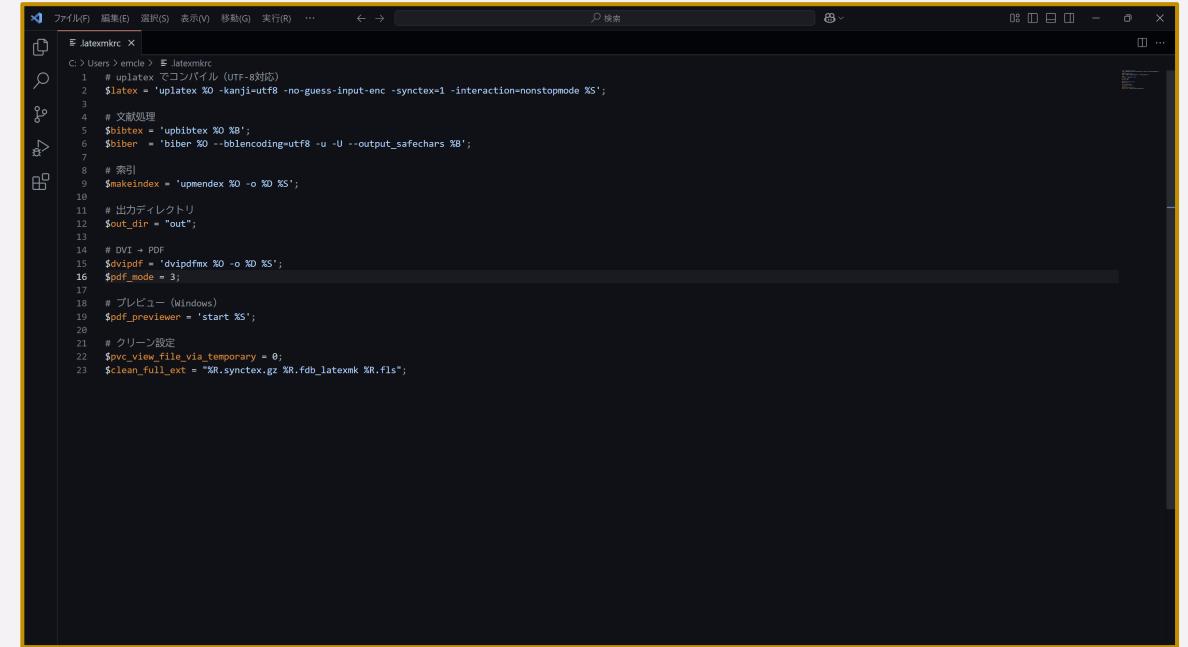
# プレビュー (Windows)
$psvc_previewer = 'start %S';

# クリーン設定
$psvc_view_file_via_temporary = 0;
$clean_full_ext = "%R.synctex.gz %R.fdb_latexmk %R.flst";
```

GitHubページにアクセス

FILE2.txt

の中身をコピーして



The screenshot shows a code editor window titled '.latexmkrc'. The code is identical to the one shown in the GitHub screenshot, indicating it has been pasted into the editor.

```
C:\>Users>emclke> .\latexmkrc
1 # uplatex でコンパイル (UTF-8対応)
2 $uplatex = 'uplatex %O -kanji=utf8 -no-guess-input-enc -synctex=1 -interaction=nonstopmode %S';
3
4 # 文献処理
5 $bibtex = 'upbibtex %O %B';
6 $biber = 'biber %O --bbencodingsutf8 -u -U --output_safechars %B';
7
8 # 索引
9 $makeindex = 'upmendex %O -o %D %S';
10
11 # 出力ディレクトリ
12 $out_dir = "out";
13
14 # DVI → PDF
15 $dvipdfm = 'dvipdfmx %O -o %D %S';
16 $pdffile_mode = 3;
17
18 # プレビュー (Windows)
19 $psvc_previewer = 'start %S';
20
21 # クリーン設定
22 $psvc_view_file_via_temporary = 0;
23 $clean_full_ext = "%R.synctex.gz %R.fdb_latexmk %R.flst";
```

.latexmkrcファイルに

ペーストして保存 [Ctrl + s]

STEP 5 | setting.jsonの設定 - LaTeXを書くために行うVS Codeの設定

34

A screenshot of the VS Code interface showing the settings editor. The title bar says 'settings.json'. The left sidebar shows a project structure with files like 'test.tex', 'setting.json', and 'settings.json'. The main area displays the 'settings.json' file content, which includes configurations for 'terminal.integrated.defaultProfile.windows', 'workbench.colorTheme', and 'python.defaultInterpreterPath'. A yellow box highlights the file path 'C: > Users > emcle > AppData > Roaming > Code > User > settings.json' in the sidebar. Below the code, a yellow box contains the text: 'すでに何かが記述されていることがあります。その場合は、[,]で繋げてその下にコピペします。' (There may already be something written. In that case, connect it with [,] and paste it below.)

1. VS Codeで設定画面を開く [Ctrl + ,]
右上のアイコンをクリックしsetting.jsonを開く
2. GitHubページにアクセス、 FILE3.txt の中身を
{この中に}コピペ

```
{  
  "terminal.integrated.defaultProfile.windows": "PowerShell",  
  "python.defaultInterpreterPath": "C:¥¥Users¥¥emcle¥¥miniconda3¥¥python.exe",  
  // 日本語文書で単語移動を使うため、助詞や読点、括弧を区切り文字として指定する  
  "editor.wordSeparators":,  
  .  
  .  
  .  
  // 生成ファイルを "out" ディレクトリに吐き出す  
  "latex-workshop.latex.outDir": "out"  
}
```

のように

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 拡張機能「LaTeX Workshop」の導入

STEP 5 .latexmkrc と setting.jsonの設定

STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

STEP 6 | ビルドテスト②

36

快適になったLaTeX執筆環境でビルドテストします。

1. GitHubページにアクセス、FILE4.txt の中身をtest.texにコピペ



The screenshot shows the LaTeX Editor interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, and Print. The main area has two tabs: 'test.tex' (active) and 'preview'. The 'test.tex' tab contains the following LaTeX code:

```
\documentclass{jsarticle}
\begin{document}
\title{はじめてのTeX}
\author{Name Wo Nyuryoku}
\maketitle
\section{TeX は意外とカンタン}
\subsection{小見出しもすぐになります}
\begin{equation}
f(x) = x^2 + 2x + 1
\end{equation}
↑ `equation` 環境で囲むだけで、こんなふうにきれいな式が出力されます。
\subsection{表も作れます（ちょっとだけめんどう）}
表も作りますが、少しだけ書き方にコツがいります。
といっても、基本形を覚えてしまえばそんなに難しくありません。
\begin{table}[htbp]
\centering
\caption{サンプル表}
\begin{tabular}{|c|c|c|}
\hline
項目 & 値1 & 値2 \\
\hline
A & 10 & 20 \\
B & 30 & 40 \\
\hline
\end{tabular}
\end{table}
↑ こんな感じで、`tabular` 環境を使って記述します。
最初はちょっと面倒に感じるかもしれません、テンプレートとして覚えてしまえば大丈夫です。
\end{document}
```

On the right side of the editor, there are two large buttons: a green triangle labeled 'ビルド' (Build) and a white square with a black border labeled 'プレビュー' (Preview). A white arrow points from the text 'マウス操作でもできます' (You can also do it with mouse operations) towards the 'ビルド' button.

2. 自動ビルド

Win: [Ctrl + alt + b]

Mac: [⌘ + ⌥(option) + b]

3. PDFプレビュー

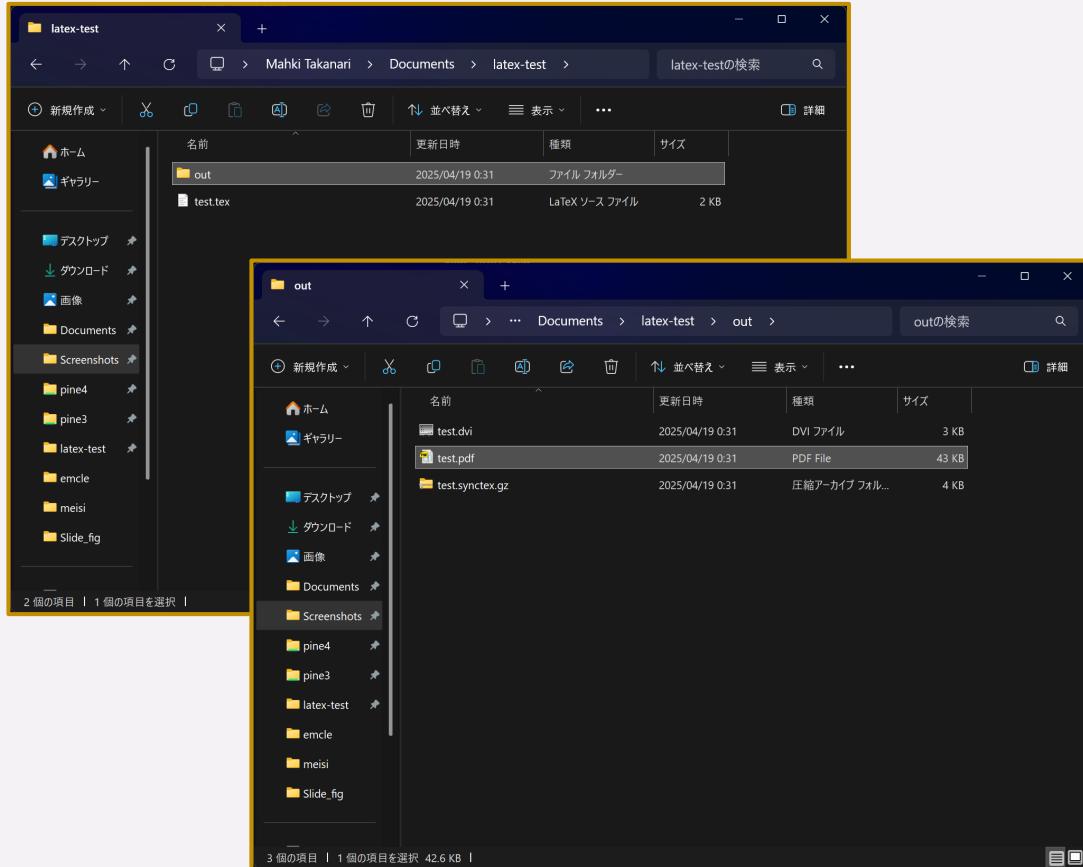
Win: [Ctrl + alt + v]

Mac: [⌘ + ⌥ + v]

一度ビルドすれば、文章を変更しても
[Ctrl (⌘) + s] でプレビューが自動更新
されるはずです。試してみてください。

STEP 6 | ビルドテスト②

37



PDFは作業ディレクトリ内に生成された
out/に入っています

■ LaTeX Workshopの便利機能

- PDFビューワー上で
[Ctrl + クリック]
[⌘ + クリック]
で対応するTeXソース箇所にジャンプ
- [Ctrl + alt + c]
[⌘ + ⌘ + c]
で生成ファイル (.auxなど) を削除

詳しくは[Wiki](#)を読んでみてください。

STEP 0 旧ver. のTeXの確認（研究室PCの人など）

STEP 1 TeX ディストリビューションのインストール

STEP 2 VS Codeのインストール（無い人）

STEP 3 ビルドテスト①

STEP 4 拡張機能「LaTeX Workshop」の導入

STEP 5 .latexmkrc と setting.jsonの設定

STEP 6 ビルドテスト②

STEP 7 トラブルシューティング

エラーが出て先に進めないことがあれば、サポートします。

TeX Liveを最小構成でインストールしているため、文書を作成しているとパッケージが無いというエラーがそのうち出るかと思います。P.26-27で説明した方法でその都度インストールすれば良いかと思いますが、それがめんどくさい場合、高成が修論で使っていたパッケージを一括インストールする方法（Win）を紹介します。

1. コマンドプロンプトを開く
2. したのコマンドを打つ（一行）

```
tlmgr install otf xparse amsmath amssymb mathtools graphicx xcolor array  
float hyperref pxjahyper siunitx cite caption lipsum
```

ただし、どんなパッケージを使っているのか自分で把握できなくなると思うので、この方法はあまりおすすめしません。自分の環境を把握できていないとエラーの対処に困ることがあるかもしれませんので、その点を理解した上で実行してください。

■ TeX

エンジンのようなもの。細かいレイアウトや
数式表現が可能だが、直接書くのは大変

■ LaTeX

TeXを使いやすくしたマクロパッケージ。
LaTeXのコマンドは、裏で自動的にTeXに
変換されて処理される。