

# platex-vscode-template の使い方 GUI 操作

Shunsuke Kimura

April 12, 2024

## 概要

Windows を使う初学者を対象に <https://github.com/kimushun1101/platex-vscode-template> をできるだけ GUI で使用する方法を示す。

ここではとりあえず方法だけ紹介するので、気になる方は各自キーワードで検索してください。

## この資料でできること

- 1 Windows 上にコンパイル速度が早い  $\text{\LaTeX}$  環境の構築
- 2  $\text{\LaTeX}$  文書作成のワークフローの習得
- 3 GitHub を用いたバージョン管理および共同作業

パスワード漏洩防止や権限の適切な設定などは各人・各組織でご管理ください。  
この資料のソースコードは MIT License としています。

<https://github.com/kimushun1101/How-to-use-platex-vscode-template>

# 目次

## ① (初回のみ) 環境構築

- WSL と Ubuntu のインストール
- L<sup>A</sup>T<sub>E</sub>X と Git のインストール
- GitHub アカウントの作成
- VS Code のインストール

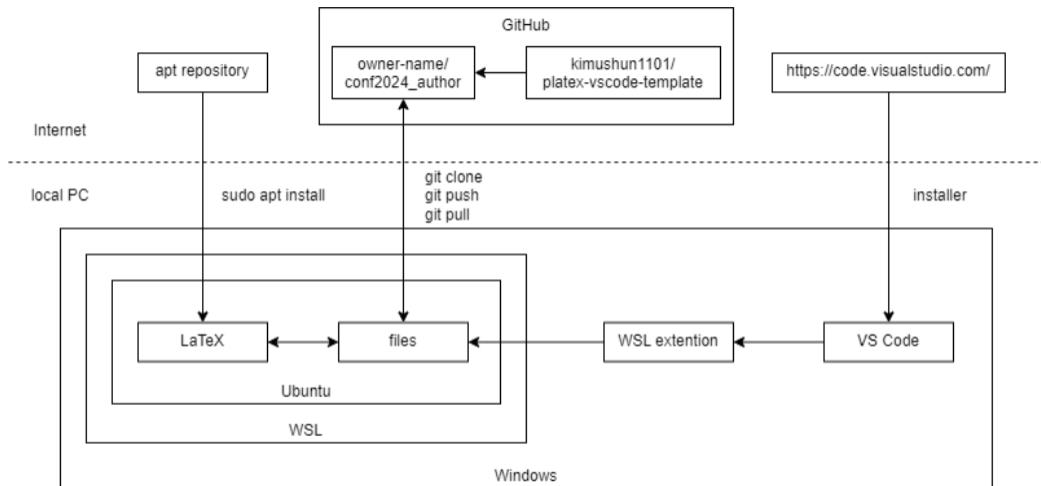
## ② 文書作成

- リモート Repository の作成
- ローカル Workspace の設定
- (初回のみ) VS Code 拡張機能のインストール
- 編集作業
- バージョンの保存

## ③ 共同作業

- (初回のみ) 組織アカウントの作成
- 変更箇所の確認
- コンフリクト (競合) の解消

# 全体構成



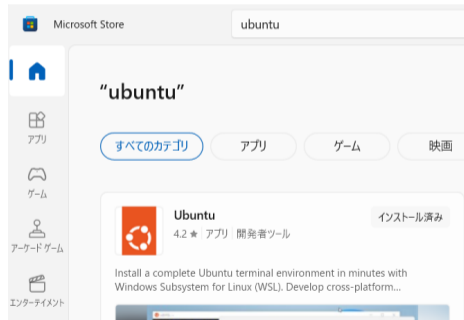
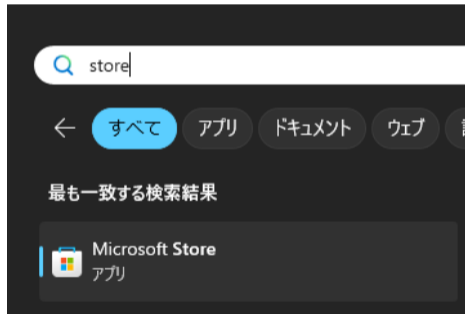
`owner-name` は適宜ご自身のユーザ名や組織名に置き換えてお考えください (後述)。

多少複雑ですが、他ソースコードの編集や管理などにも流用できる技術で構成しております。

# (初回のみ) 環境構築 1: WSL と Ubuntu のインストール

Microsoft Store から Ubuntu をインストールする。

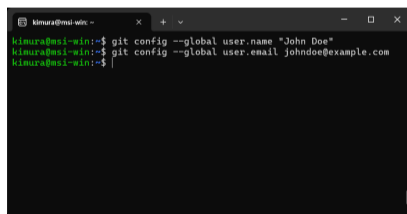
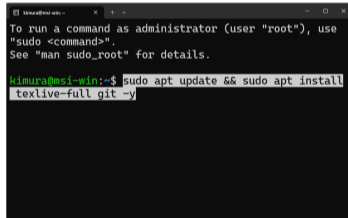
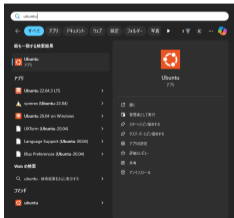
- ① スタートメニューから「Store」と検索すると「Microsoft Store」が見つかる。
- ② Microsoft Store の検索窓に「Ubuntu<sup>1</sup>」と検索する。



<sup>1</sup>もし WSL を  $\text{\LaTeX}$  しか使用しない場合には「Alpine」の方が次ページのバグを踏まないのが良いかも

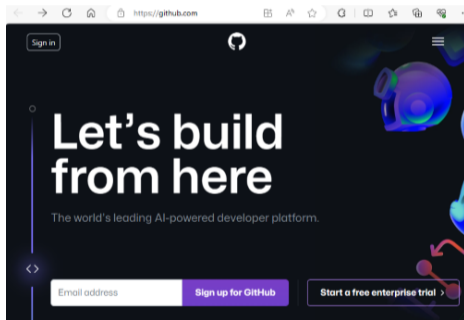
# (初回のみ) 環境構築 2: L<sup>A</sup>T<sub>E</sub>X と Git のインストール

- 1 Ubuntu を実行. ユーザ名とパスワードを設定する. パスワードは \*\*\* などで表示されないことに注意.
- 2 `sudo apt update && sudo apt install texlive-full git -y` コマンドを入力. パスワードを聞かれるので設定したパスワードを入力. Ubuntu ではインストール中に固まってしまうバグがあるが Enter を何回か押せばとりあえず進む [1].
- 3 `git config --global user.name "John Doe"` を 自身の名前 (匿名可) に変更して `git config --global user.email johndoe@example.com` を 自身のアドレスに変更して それぞれ入力 [2].



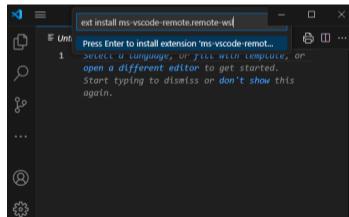
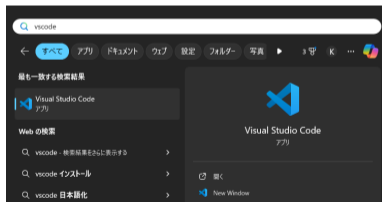
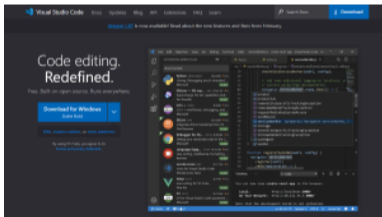
## (初回のみ) 環境構築 3: GitHub アカウントの作成

- 1 ブラウザで <https://github.com/> にアクセスして、[Sign up for GitHub](#) から、ユーザ名や E メールアドレス、パスワードなどを設定 [3]。以降ログインした状態とする。



# (初回のみ) 環境構築 4: VS Code のインストール

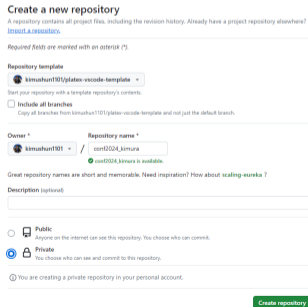
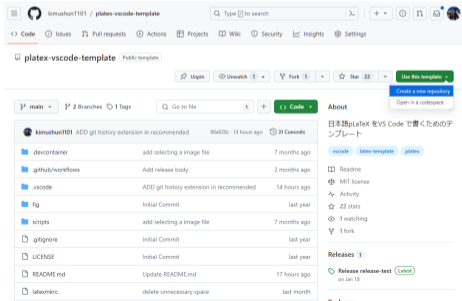
- 1 ブラウザで <https://code.visualstudio.com/> を開き, **Download for Windows** をクリックしてダウンロードできるインストーラーを実行.
- 2 インストール後に VS Code を実行して **Ctrl + P** で現れるテキストボックスに `ext install ms-vscode-remote.remote-wsl` を入力して WSL Extension を install.





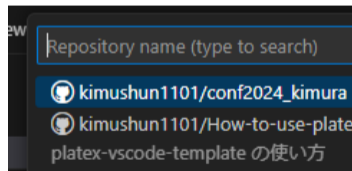
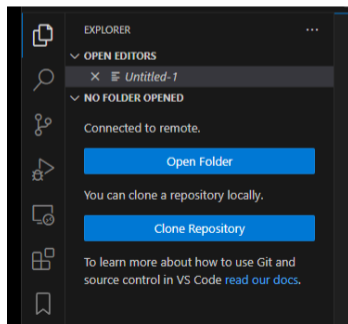
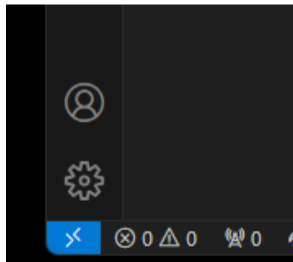
# 文書作成 1: リモート Repository の作成

- 1 <https://github.com/kimushun1101/platex-vscode-template> にアクセス。
- 2 `Use this template` から `create a new repository` を選択
- 3 推奨設定は以下の通り（あとから変更も可能）
  - 共有したい場合には Owner を組織アカウントに変更（後述）。
  - Repository name は、ディレクトリ名（フォルダ名）になります。投稿先や筆頭著者名、タイトル名の一部など識別できるものを推奨。例として `conf2024_kimura` としたとして進める。
  - Private Repository にチェックを入れる。



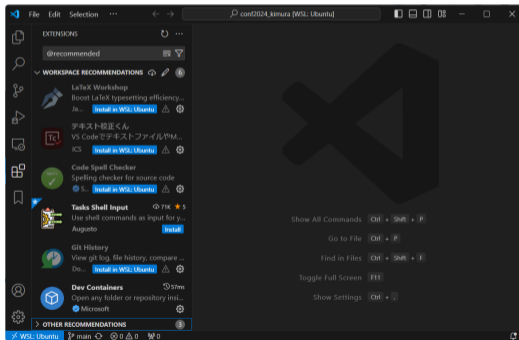
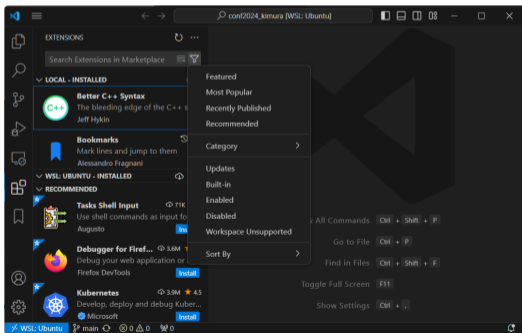
## 文書作成 2: ローカル Workspace の設定

- 1 VS Code で New Window を開く. ショートカットは `Ctrl + Shif + N`.
- 2 左下の `><` から `Connect to WSL` をクリック.
- 3 `Clone Repository` をクリック. GitHub へのサインインを求められたら行う [4].
- 4 `Clone from GitHub` から今回作成した `conf2024_kimura` を選択.
- 5 ポップアップが出たら `Open` を選択.



# 文書作成 3: (初回のみ) VS Code 拡張機能のインストール

- 1 VS Code 画面左にある Extensions を開く. ショートカットは `Ctrl + Shif + X`.
- 2 検索ボックス右の Filter Extensions をクリックして, `recommended` を選択して出てきたものをすべてインストールする.



## 文書作成 4: 編集作業 (サンプルの確認)

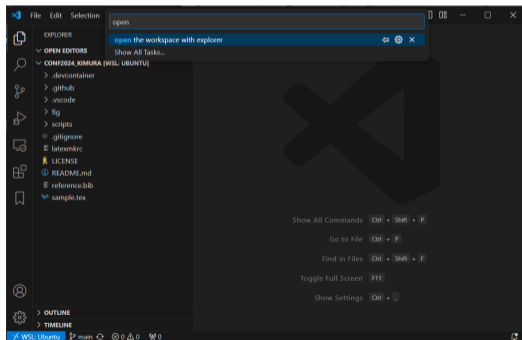
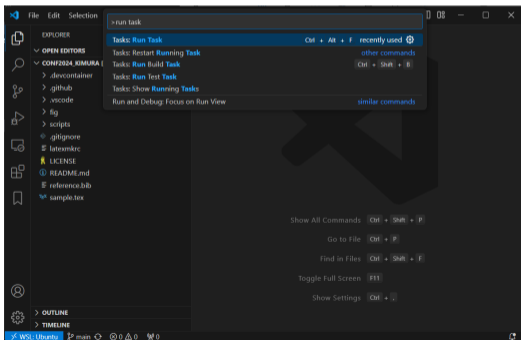
- 1 VS Code 画面左にある Explore を開く. ショートカットキーは `Ctrl + Shift + E`.
- 2 `sample.tex` をクリックして開く.
- 3 `Ctrl + Alt + B` でビルド. ファイル保存時もオートビルドされる.
- 4 `Ctrl + Alt + V` でコンパイル済み PDF を表示.

その他のショートカットは以下の通り.

ショートカット	機能
<code>Ctrl + Alt + J</code>	PDF の該当の場所にジャンプ
PDF を <code>Ctrl</code> + クリック	<code>.tex</code> の該当の行にジャンプ
<code>Ctrl + Alt + M</code>	数式プレビューの表示
<code>Ctrl + J</code> (PROBLEMS タブ)	エラーやワーニングの確認
<code>F8</code>	エラーやワーニングへジャンプ

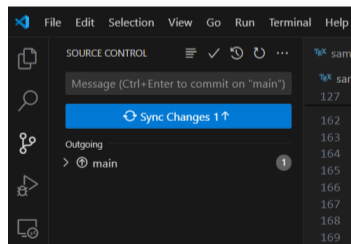
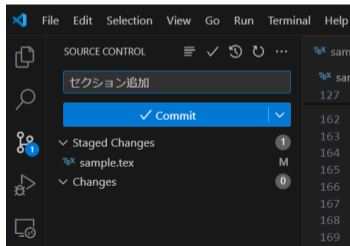
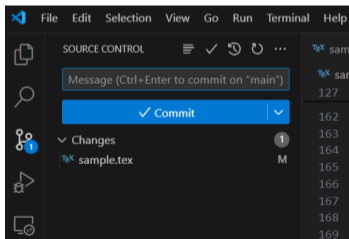
# 文書作成 4: 編集作業 (テンプレートの導入)

- 1 学会などから投稿先のテンプレートファイル入手. 圧縮されている場合は展開する.
- 2 `Ctrl + Shift + P` から `Tasks: Run Task` を選択し, つづけて `Open the workspace with explorer` を選択. フォルダにファイルをコピーする.
- 3 ビルドしたい `.tex` ファイルを開きビルドする. ファイル名は `sample.tex` でなくても有効.



## 文書作成 5: バージョンの保存

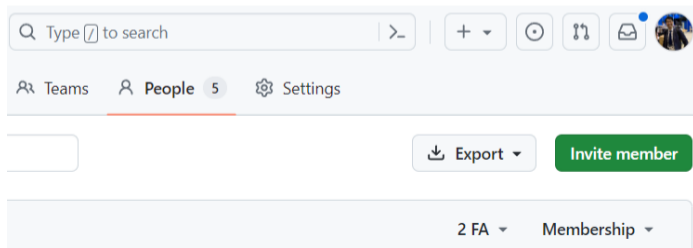
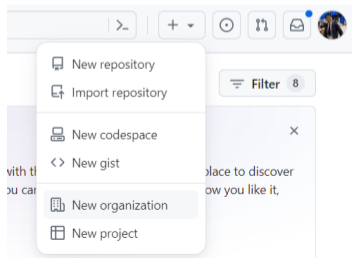
- 1 VS Code 画面左にある Source Control を開く。ショートカットキーは `Ctrl + Shift + G`。
- 2 Changes から保存したいファイルを `+` で追加。Staged Changes に乗せる。
- 3 Message に今回の変更内容を記入。
- 4 `Commit` の後、`Sync Changes` でアップロード。
- 5 ブラウザで GitHub repository ページを更新するとアップロードを確認できる。



# 共同作業 1: (初回のみ) 組織アカウントの作成

組織アカウントが Owner である Private Repository は、組織アカウントに紐づけられた個人アカウントのみ閲覧することができる。

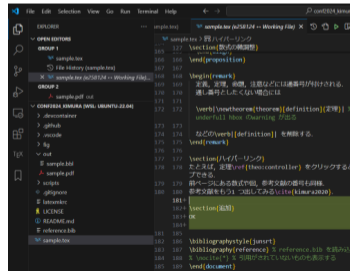
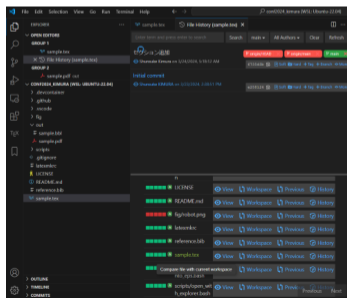
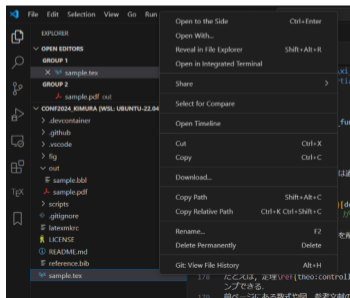
- 1 <https://github.com/> の左上にある **+** から **New organization** をクリック。
- 2 とりあえずは Free プランの **Create a free organization** を選択する。
- 3 Organization name や E メールアドレスなどを入力して作成。
- 4 <https://github.com/organization-name> を **organization-name** を置き換えて アクセス。
- 5 People タブから **Invite member** をクリックして、メンバーを追加。



# 共同作業 2: 変更箇所の確認

バージョン間の差分を確認できる。

- 1 Source Control から **Sync Changes** をクリックすることで、他の人が行った変更を取り込む。
- 2 VS Code で Explore を開く。ショートカットキーは **Ctrl + Shift + E**。
- 3 履歴をみたいファイルを右クリックして **Git: View file history** を選択。ショートカットは **Alt + H**。
- 4 対象のコミットを選択して、対象のファイルの **Workspace** をクリック。
- 5 対象のファイルの **View** をクリックすれば、そのコミット時点での内容を確認できる。

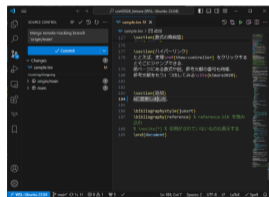
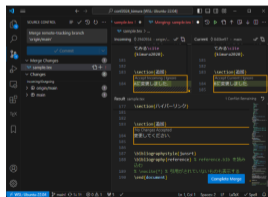
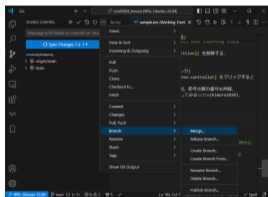
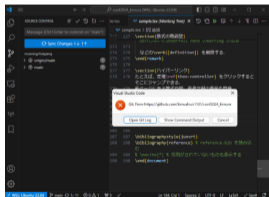




## 共同作業 3: コンフリクト (競合) の解消

自身の変更をアップロードする前に、他の人の変更がアップロードされていた場合、手元でコンフリクトを解決したコミットを作成してからアップロードする。

- 1 バージョンの保存をした際にポップアップが出た場合、**Sync Changes** がダウンロード・アップロード両方ある状態であることが確認できる。
- 2 SOURCE CONTROL の **...** から **Branch**、**Merge** をクリックし、**origin/main** を選択。
- 3 **Accept Current Change** や **Accept Incoming Change** などをクリックして、適宜追加変更する。
- 4 編集したファイルを **+** で追加、**Commit** の後、**Sync Changes** でアップロード。



まじめにバージョン管理をするのであれば git branch や git tag も合わせて調べてみてください [5].

- [1] Pregenerating ConTeXt MarkIV format. This may take some time... takes forever  
<https://askubuntu.com/questions/956006/pregenerating-context-markiv-format-this-may-take-some-time-takes-forever>
- [2] Git Book 1.6 Getting Started - First-Time Git Setup  
<https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
- [3] GitHub アカウントの概要  
<https://docs.github.com/ja/get-started/onboarding/getting-started-with-your-github-account>
- [4] GitHub への認証方法について  
<https://docs.github.com/ja/authentication/keeping-your-account-and-data-secure/about-authentication-to-github>
- [5] Using Git source control in VS Code  
<https://code.visualstudio.com/docs/sourcecontrol/overview>