

## Part I

# wslでtexliveをinstallする

## 1 TeXLive をインストール

Ubuntu/Debian 系の場合、ターミナルで以下を実行する。

```
1 sudo apt update
2 sudo apt install texlive-full
```

Windows の場合

1. TeXLive の公式サイトからインストーラーをダウンロードする。
2. インストーラーを実行し、指示にしたがってインストールする。

macOS の場合、以下を実行する。

```
1 brew install --cask mactex
```

## 2 VS Code の拡張機能をインストール

VS Code で LaTeX を快適に使うために、以下の拡張機能をインストールする。

1. LaTeX Workshop をインストールする
2. VS Code を開く
3. 左側の拡張機能アイコンをクリック
4. 検索バーに「LaTeX Workshop」と入力
5. 「LaTeX Workshop」を選択し、「インストール」をクリック

### 3 TeXLive と VS Code の設定

LaTeX Workshop 拡張機能を適切に動作させるために設定を確認します。

1. 設定ファイルを編集

以下の内容を ‘Settings (JSON)’ に追加します。

```
1  "latex-workshop.latex.toolchain": [  
2      {  
3          "command": "pdflatex",  
4          "args": [  
5              "-synctex=1",  
6              "-interaction=nonstopmode",  
7              "-file-line-error",  
8              "%DOC%"  
9          ]  
10     }  
11 ],  
12 "latex-workshop.latex.outDir": "./out"
```

2. 必要に応じて環境変数を確認

以下を実行して、PATH が正しいことを確認します。

```
1  echo $PATH
```

### 4 動作確認

1. ‘tex’ファイルを作成

```
1  \documentclass{article}  
2  \begin{document}  
3  Hello, LaTeX!  
4  \end{document}
```

2. ファイルを保存し、自動で PDF が生成されるか確認します。

## Part II

# rgt レポジトリをクローンし、rgt.sty を texmf-home に設置する

## 5 レポジトリをクローン

以下を実行してレポジトリをクローンします。

```
1 git clone https://github.com/username/rgt.git
```

## 6 クローン後の確認

1. ディレクトリに移動し、ファイル一覧を確認します。

```
1 cd rgt
2 ls
```

2. 出力された結果からどのディレクトリにクローンされているかがわかります。

```
3.
1 README.md doc example
```

## 7 rgt.sty を texmf-home に設置

以下を実行して、‘texmf-home’にパスが通っていることを確認します。

```
1 mktexlsr
2 kpsewhich rgt.sty
```

## Part III

# 多重texmfツリー

## 8 多重texmfツリーとは

異なる種類の TeX パッケージやフォントなどを整理し、管理するためのディレクトリ構造のこと。

## 9 基本的な texmf ツリーの構成

### 1. メインの texmf ツリー

- TeX Live や MiKTeX などの TeX ディストリビューションに含まれる標準のツリー
- グローバルなパッケージやフォントがインストールされる

### 2. ローカル texmf ツリー

- ユーザが追加のパッケージやフォントをインストールするためのツリー
- システムのメインツリーに影響を与えることなく、個別のカスタマイズが可能

### 3. ホーム texmf ツリー

- 個々のユーザーのホームディレクトリ内に設定されるツリー
- ユーザー固有の設定やパッケージを保持

## 10 多重 texmf ツリーの利点

### 1. 柔軟性

- 異なるユーザーが異なる設定やパッケージを使用できるため、複数人での作業が効率的に行える

### 2. 安全性

- システムワイドな設定を変更することなく、ユーザーが自由にパッケージを追加・更新できる

### 3. 整理整頓

- パッケージを目的別に分類することで、管理がしやすくなる