

目次

第 1 章	はじめに	2
第 2 章	環境構築と初期設定	3
2.1	L ^A T _E X を使うのに必要なもの	3
2.2	T _E XLive の導入	3
2.3	VSCo ^d e の導入	5
2.4	VSCo ^d e の設定	6
第 3 章	L ^A T _E X の基本文法	8
3.1	L ^A T _E X 文書の基本	8

第 1 章

はじめに

お手に取って頂きありがとうございます。

みなさんは $\text{\LaTeX}^{\ast 1}$ というソフトウェアをご存知でしょうか？ \LaTeX とは $\text{\TeX}^{\ast 2}$ を元に開発された文書作成ソフトウェアで、編集している画面が出力として得られる Microsoft Word などのソフトウェアとは対照的に \LaTeX の文書はプログラミング言語のような形で命令と文章を記述し、タイプセットと呼ばれるコンパイルを行うことで PDF 形式での出力を得られる、という形の文書作成システムです。

このような形式は一見面倒に思えますが、自動で段落や目次の生成を行えたり、強力な図形描画機能を備えている点から、レポートはもちろん論文の執筆で威力を発揮します。また美しい文書を作成できることでも評価が高く、 \LaTeX を用いた書籍も多数出版されています。もちろん本書も \LaTeX で作成されています。

本書では、 \LaTeX の基本的機能から、レポート執筆に便利なグラフの生成や回路図の生成が可能になる拡張機能の使用法の解説、またソースファイルの差分管理など運用面の内容も交えて解説します。

本書が読者のレポート執筆の一助となれば幸いです。

^{\ast 1} ラテック/ラテフ、英語圏ではレイテックとも読まれる

^{\ast 2} テック/テフと読む、テックスは誤り

第 2 章

環境構築と初期設定

L^AT_EX を使うにあたり最も大きな障壁とされるのが環境構築^{*1}とされています。確かに Microsoft Word などと比べれば導入は少々煩雑ではありますが、多くの方々の尽力により今ではとても簡単になっているので、身構えることはありません。

2.1 L^AT_EX を使うのに必要なもの

まず、L^AT_EX のソフトそのものが必要になりますが、L^AT_EX は単体のソフトウェアではなく、多くの関連ソフトの集合体です。それらを一つ一つ導入していくのはとても手間がかかるので、L^AT_EX では関連するソフトをひとまとまりにした状態で配布するディストリビューションという形態がとられています。現在配布されているディストリビューションにも数種類あるのですが、現在最もポピュラーな T_EXLive というディストリビューションを今回使用します。

また L^AT_EX は、マークアップ言語と呼ばれるプログラミング言語のような形で文章の構造を指定します。そのため L^AT_EX を使うためには L^AT_EX 本体ソフトウェア以外にもマークアップ言語を記述するためのテキストエディタが必要になります。T_EXLive にも一応 T_EXWorks というテキストエディタが同梱されているのですが、お世辞にもモダンとは言えません。ですので、今現在 L^AT_EX に限らず多くのプログラミング言語の開発環境に用いられている Visual Studio Code というエディタを使用します。

今回インストールするソフトウェアは以下の 2 つとなります。

L^AT_EX ディストリビューション T_EXLive
テキストエディタ Visual Studio Code

これらのソフトウェアの導入手順を以下にて解説します。

2.2 T_EXLive の導入

まず最初に T_EXLive を導入します。

^{*1} そのソフトを使える環境を整えること

2.2.1 Windows

以下のページよりネットワークインストーラをダウンロードします。

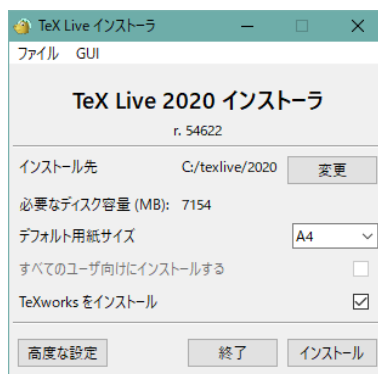
```
https://www.tug.org/texlive/acquire-netinstall.html
```

このページを開いて、`install-tl-windows.exe` をダウンロードします。

ダウンロードしたファイルを起動すると展開用のソフトが起動するので、`install` を選択し次に進み、`install` ボタンを押下します。すると右のようなインストーラが起動します。ここで様々なインストールの設定が行えますが、たいいていの場合変更は必要ありません。インストール先もこの設定で変更が可能ですが、本書では変更しないという前提で解説をします。

インストールボタンを押下するとインストールが開始されます。かなり時間がかかるので放置しておきます。

終わったらインストーラを閉じてインストールは完了です。



2.2.2 Linux

Debian, Ubuntu など

パッケージ管理ソフトを使用するのでまず最初に更新を行います。以下のようにしてパッケージとリストを更新します。

```
$ sudo apt update
$ sudo apt upgrade
```

更新したら、以下のようにして TeX Live と関連ソフトウェアをインストールします。

```
$ sudo apt install texlive-lang-japanese ghostscript perl
$ sudo apt install evince poppler-utils poppler-data
$ sudo apt install texlive-fonts-recommended texlive-fonts-extra
```

以上で完了です。

Arch Linux など

Ubuntu などと同様にパッケージリストを更新します。

```
$ sudo pacman -Syu
```

更新したら、同様に以下のようにして TeX Live と関連ソフトウェアをインストールします。

```
$ sudo pacman -S texlive-langjapanese texlive-most
$ sudo pacman -S ghostscript perl
$ sudo pacman -S evince poppler-utils poppler-data
```

以上で完了です。

2.3 VSCode の導入

次に Microsoft Visual Studio Code(以後 VSCode と省略) を導入します。

2.3.1 Windows

以下のページよりインストーラをダウンロードします。

```
https://code.visualstudio.com/
```

インストーラを起動して、指示にしたがってインストールを進めます。

以下のようなチェックボックスのある画面が出たら、

- 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^{*1}
- 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^{*1}
- PATH への追加

のチェックボックスにチェックを入れて次へを押下します。その後は指示にしたがってインストールを進めたら完了です。

2.3.2 Linux

Debian,Ubuntu など

以下のページより.deb 形式のファイルをダウンロードします。^{*2}

```
https://code.visualstudio.com/
```

以下のコマンドを実行してダウンロードした.deb 形式のファイルからインストールします。ファイルパスはダウンロードしたファイル名に適宜変更します。

^{*1} この項目は必須ではありませんが便利なのでおすすめです

^{*2} コマンドラインのみでのインストールも可能ですがここでは省略します。

詳細は <https://code.visualstudio.com/docs/setup/linux> を参照してください。

```
$ sudo apt install ./<file>.deb
```

以上で完了です。

Arch Linux など

以下のコマンドを実行してパッケージ管理ソフト経由でインストールします。^{*3}

```
$ sudo pacman -S code
```

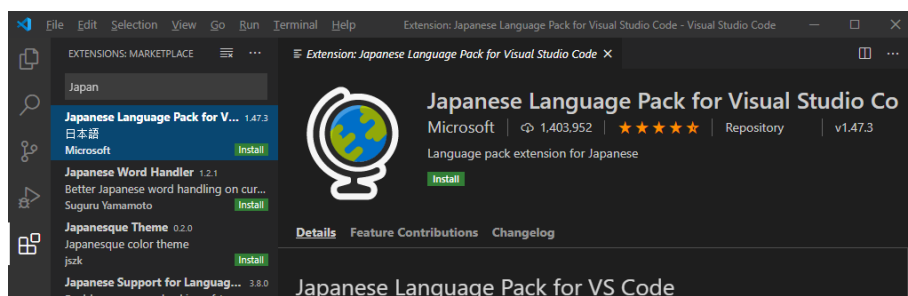
以上で完了です。

2.4 VSCode の設定

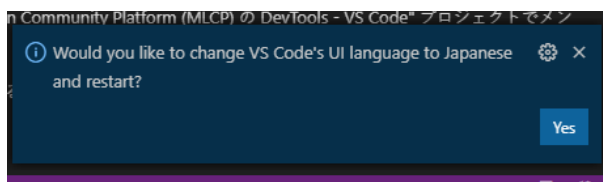
2.4.1 日本語環境のセットアップ

前項で VSCode のインストールを行いました。インストールしたままの状態だと \LaTeX の執筆があまりしやすいとは言えません。そのため本項では、 \LaTeX の執筆がしやすいように VSCode の設定を変更していきます。

インストールしたままの状態では言語環境が英語なので、日本語の環境に設定します。VSCode を起動し、キーボードで **Ctrl** + **Shift** + **X** を押下します。拡張機能のタブが開かれるので、検索バーに Japan と入力します。すると候補に Japanese Language Pack for Visual Studio Code が表示されるので、インストールのボタンを押下します。



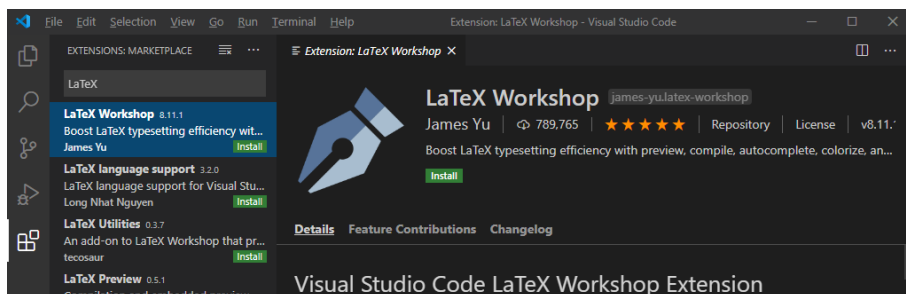
インストールが完了すると以下のようにリロードを要求されるので、Yes を押下してリロードを行い日本語環境の設定は完了です。



^{*3} このリポジトリは OSS 版であり、Microsoft 公式のリポジトリではないようです。使用に関して全く問題はありますが、公式のリポジトリからインストールする場合は <https://code.visualstudio.com/docs/setup/linux> を参照してください。

2.4.2 LaTeX Workshop の導入

次に \LaTeX の執筆を支援する拡張機能である LaTeX Workshop を導入します。先ほどと同様に **Ctrl** + **Shift** + **X** を押下し拡張機能のタブを開き、検索バーに LaTeX と入力します。候補の中から LaTeX Workshop を選択し、インストールを押下します。



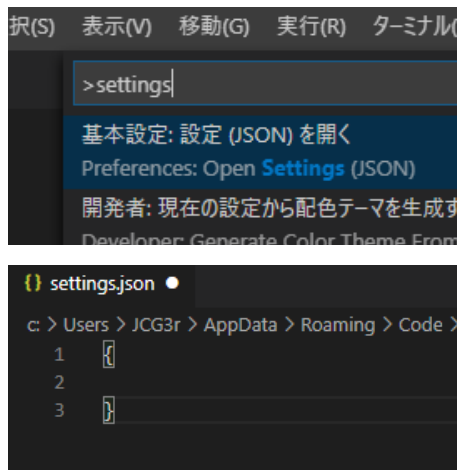
しばらくするとインストールが終わった旨の表示がされ、導入は完了となります。

2.4.3 VSCode の設定

執筆に当たり必要な拡張機能は入れ終わったので、設定をしていきます。まず、**Ctrl** + **Shift** + **P** を押下して、settings と入力し、候補に出てきた『基本設定: 設定 (JSON) を開く』を選択して開きます。すると図のように "settings.json" が開かれることを確認します。次に、以下のページから設定ファイルをコピーします。

<https://gist.github.com/>

コピーした設定ファイルを先ほど開いた "settings.json" の `{ }` の中に貼り付けます。貼り付けた後、**Ctrl** + **S** で保存して設定は完了です。



2.4.4 動作確認

これまでの作業で正しく環境構築と初期設定が行われているか、実際に \LaTeX 文書を作ってみて確認してみましょう。

第 3 章

L^AT_EX の基本文法

前章で L^AT_EX の環境構築と初期設定が終わりました。少々長かったですが、これで L^AT_EX を使用する準備は万端です。本章では、構築した環境を利用して L^AT_EX の基本的な機能を使った文書を作っていきます。

3.1 L^AT_EX 文書の基本

まず、基本的な L^AT_EX 文書の構成について解説します。L^AT_EX 文書の基本的な例を以下に示します。

```

1 \documentclass[uplatex,dvipdfmx]{jsarticle}
2
3 \title{テスト文書}
4 \author{わたし}
5 \date{\today}
6
7 \begin{document}
8     \maketitle
9
10    \section{セクション}
11        セクションです。
12
13    \subsection{サブセクション}
14        サブセクションです。
15
16 \end{document}

```

L^AT_EX の構文では、大きく分けて 2 種類の記述があります。1 行目や 3 行目のように `\` で始まるコマンドという記述と、11 行目や 14 行目のように `\` で始まらない文章があります。文章は書いた内容がそのまま出力させる指示になりますが、コマンドの場合は L^AT_EX ソフトウェアに特殊な指示を与えます。

また L^AT_EX 文書は、その文書の設定について記述するプリアンブルと、文書そのものを記述する本文の 2 つで構成されます。たとえばこの文書では、ハイライトされている 1 行目から 6 行目がプリアンブルにあたります。プリアンブルでは通常の文章は記述できません。ハイラ

イトされていない`\begin{document}`で始まり`\end{document}`で終わる部分が本文にあたります。

3.1.1 プリアンブルの内容

`\documentclass`ではその文書のタイプセット手順、言語設定やレイアウトなどについて設定します。今回の場合日本語環境で最も使われる L^AT_EX エンジンの uplatex を使用するよう指定し、後処理に dvipdfmx というソフトを使用するように指定しています。また文書のレイアウトを日本語環境レポート向けの jsarticle に指定しています。このコマンドは L^AT_EX 文書において必ず記述する必要があります。

`\title`では文書のタイトルを、`\author`では文書の作成者を、`\date`では文書の作成日など、ここではタイトルに記述する内容について設定しています。しかし、タイトルの出力自体はこの記述ではされません。本文中の`\maketitle`を記述することによってタイトルが出力されます。

3.1.2 本文の内容

`\maketitle`ではプリアンブルで設定したタイトル・作成者・作成日を出力します。`\maketitle`を記述しないとタイトルが作成されず、いきなり本文から始めることができます。

`\section`では文書の節を出力します。このコマンドは記述した箇所から次に`\section`が記述されるまでの文章がその節の区間となります。つまりこの文書では 10 行目から 12 行目までに記述された内容が第 1 節となります。

`\subsection`では文書の小節を出力します。`\section`より 1 つレベルが低くなり、第 1 節の範囲内で`\subsection`を使用すると第 1.1 節といったように出力されます。同様のコマンドに`\subsubsection`なども存在します。