

VSCode による L^AT_EX 環境構築

2026 年 2 月 17 日

本稿は全くの初心者が文書作成ツールである L^AT_EX の環境構築を完了することを目的として作成する。まずはじめに TexWorks^{†1} をインストールし、その後 VSCode^{†2} での環境構築を行う。

1 L^AT_EX の環境構築

1.1 L^AT_EX とは (既に知っている人は次節へ)

L^AT_EX (ラテフ) について解説する前にその元となっているソフトである T_EX (テフ) について解説する。T_EX とはオープンソースの組版ソフトである。組版は印刷用語で、活字を組んで版を作ることを意味する。T_EX は、コンピュータでテキストと図版をうまく配置して版にあたるもの (pdf 等のファイル) を出力するためのソフトである。特に数式の組版について定評があり、数式をテキスト形式で表す際の事実上の標準となっている。例えば以下のように数式を記述することができる。

$$y = \frac{m}{k} \left\{ \left(v_0 \sin \theta + \frac{m}{kg} \right) \left(1 - e^{\frac{k}{m}t} \right) - gt \right\} + y_0$$

次に、L^AT_EX について解説する。L^AT_EX とは機能強化された T_EX である。L^AT_EX になったことで、文書の論理的な構造と視覚的なレイアウトを分けて考えることができるようになった。たとえば、「はじめに」という ^{セクション} 節の見出しがあれば、文書ファイルには

```
\section{はじめに}
```

のように書いておく。この `\section{はじめに}` という命令が、紙面上のデザイン、例えば「14 ポイントのゴシック体で左寄せ、前後の空白はそれぞれ何ミリを標準とし…」というレイアウトに対応するといったことは、様式、版ごとに別ファイル (クラスファイル) に記述されている。標準のクラスファイルのデザインが気に入らないなら、自由に変更できる。クラスファイルだけ変更すれば同じ文書ファイルでも違ったレイアウトで出力できる。これが、文書の論理的な構造と視覚的なレイアウトを分けて考えることができるということである。

さて、L^AT_EX にも様々な種類があるが、本稿で扱うのは LuaL^AT_EX というものである。これ以前に日本語の扱いに特化した L^AT_EX である pL^AT_EX という L^AT_EX があった。これをさらに改良したものが LuaL^AT_EX だ。

1.2 TexWorks のインストール (既にインストールされている人は次節へ)

L^AT_EX を利用するには Windows なら、TexWorks という L^AT_EX 編集ソフトをインストールする必要がある。TexWorks は以下のリンク先の `install-tl-windows.exe` をクリックすることでインストール可能である。

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

ダウンロード^{†3} が完了したらインストーラからインストール^{†4} する必要がある。これには通常かなりの時間がかかるので辛抱強く待っていただきたい。インストールが完了したら TexWorks の準備は完了である。

^{†1} 公式の L^AT_EX 編集ソフトであるが、最低限の機能しか備えておらず、使い勝手が悪いのでこれでの編集は避けたい。

^{†2} Visual Studio Code のことで、Microsoft が提供しているプログラミング用のコードエディタである。様々な機能を有しており TexWorks に比べて使い勝手がよく、多くの人がこれで T_EX の編集を行う。

^{†3} インターネットからファイルを取得することをダウンロードと呼ぶ。

^{†4} インターネットから取得したものを自分のコンピュータで使えるようにすることをインストールと呼ぶ。

2 VSCode の環境構築

2.1 VSCode とは (既に知っている人は次節へ)

VSCode とは Visual Studio Code のことで Microsoft が提供しているプログラミング用のコードエディタである。様々な拡張機能が存在しており、それらをインストールすることで TexWorks よりも遥かに便利に $\text{T}_{\text{E}}\text{X}$ の編集を行うことができる。例えば、コマンドの入力補完や出力したい記号から $\text{T}_{\text{E}}\text{X}$ コマンドの入力等々挙げれば枚挙にいとまがない。

2.2 VSCode のインストール (既にインストールされている人は次節へ)

さて、VSCode は以下のリンクからダウンロードすることができる。

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

ダウンロードが完了したらインストーラーに従ってインストールを行う。インストールが完了したら日本語に変更したい人は Extension のタブから Japanese Package をインストールするとよい。

2.3 VSCode のセットアップ

はじめに Extension(拡張機能) から LaTeX Workshop^{†5}, indent-colorizer^{†6}等をインストールする。

次に $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Workshop の設定を変更する。冒頭に述べた通り $\text{T}_{\text{E}}\text{X}$ には様々な種類があるため、この設定によって $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ がインストールされるようにする。まず Windows の人は “Ctrl”+“,” を押すことによって VSCode の設定を開く。次に右上にある四角に折り返しの矢印が書いてあるマークを押すことによって settings.json を起動する。これは VSCode の設定が記述されているファイルで、ここに LaTeX Workshop の設定を追記する。以下に示すコードをコピーする。このとき既に記述されている人はその記述の最後に “,” を追記してから以下をコピーする。そうでない人はそのままコピーすればよい。なお、pdf から選択してコピーするのではなく、同じディレクトリに格納してある settings.json ファイルから直接コピーすればよい。

Listing 1 VSCode の settings.json に追記

```
1 [
2   "latex-workshop.latex.outDir": "out",
3   "latex-workshop.latex.autoClean.run": "onBuilt",
4   "latex-workshop.latex.autoBuild.run": "onSave",
5   "latex-workshop.view.pdf.viewer": "tab",
6   "latex-workshop.latex.recipes": [
7     {
8       "name": "lualatexmk",
9       "tools": ["optimized_lualatexmk"]
10    }
11  ],
12  "latex-workshop.latex.tools": [
13    {
14      "name": "optimized_lualatexmk",
15      "command": "latexmk",
16      "args": [
17        "-lualatex",
18        "-synctex=1",
19        "-interaction=nonstopmode",
20        "-file-line-error",
21        "-outdir=%OUTDIR%",
22        "-auxdir=%OUTDIR%",
```

^{†5} この拡張機能を使って TexWorks を裏側から呼び出し機能のみ利用し、VSCode で動かすことが可能になる。

^{†6} インデントが色付けされるため見やすくなる。必須ではないが便利な機能。

```

23     "-cd",
24     "-e",
25     "$success_cmd = 'echo compile succeeded';",
26     "%DOC%"
27 ],
28 "env": {
29     "LUAOPT": "--luatex-cache",
30     "TEXMFVAR": "%OUTDIR%/texmf-var"
31 }
32 }
33 ]
34 ]

```

これで一通りの設定が完了したので次章で動作を確認を行う。

3 動作確認

ここまでで環境構築が完了したので最後に動作確認を行う。以下のサンプルコードをコピーしてうまく動作すれば設定終了である。以下のサンプルコードで示している`\documentclass`とは、作成する文書の種類を示しており、今回は一般的なA4文書である`ltjsarticle`を使用している。また、3行目以降の`\usepackage{...}`では、作成する文書内で使用するライブラリを宣言している。LaTeXには様々なライブラリが用意されているため、目的に応じて様々なライブラリを使用するとよい。他にも、`\usepackage{newunicodechar}`を使用することで、句読点とコンマピリオドを自動で置換することなども可能である。ここまでの記述を**プリアンプル**といい、文書の設定部分に値する。本文は`\begin{document}`と`\end{document}`の中に書く必要がある。タイトルの出力は、`\title{タイトル}`、`\author{著者}`、`\date{\today}`のように内容を設定し^{†7}、`\maketitle`とすることで設定したタイトルを出力することができる。

動作させるには、まず“Ctrl” + “s”を入力することで、ファイルを保存する。このとき、ファイルの拡張子^{†8}を`.tex`にする必要がある。保存が完了したらもう一度“Ctrl” + “s”を入力することでコンパイル^{†9}が始まる。画面左下のぐるぐるがチェックマークに変わったら、“Ctrl” + “Alt” + “v”を入力することで作成したpdfファイルを表示させることができる。

Listing 2 サンプルコード

```

1 \documentclass[] {ltjsarticle}
2 \usepackage{graphicx} %図形の利用
3 \usepackage{bm} %ボールドの利用
4 \usepackage{newtxtext} %本文にTimes系フォントの利用
5 \usepackage{ascmac}%枠線で囲めるようになる
6 \usepackage[margin=15mm]{geometry}%余白を小さくする
7
8
9 %本文開始
10 \begin{document}
11 %タイトル
12 \title{動作確認}
13 \author{朕}
14 \date{\today}
15 \maketitle
16
17 朕の文章がうまく動作しておる。
18 \end{document}

```

^{†7} `\today` とすることで、作成日を自動で入力することができる。

^{†8} ファイルのほげほげ.pptx などの. なんちゃらの部分を拡張子という

^{†9} ここでは LaTeX のコードの pdf 化を意味する。

参考文献

- [1] 奥村, 黒木 (2020), 『改訂第 8 版 L^AT_EX 2_ε 美文書作成入門』, 技術評論社