VSCode による LaTeX 環境構築

2026 年 2 月 26 日

本稿は全くの初心者が文書作成ツールである LaTeX 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．

目次

1 LaTeX の環境構築

1.1 LaTeX とは (既に知っている人は次節へ)

LaTeX(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 TeX(テフ) について解説する．TeX 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．TeX 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

[†1] 公式の LaTeX 編集ソフトであるが，最低限の機能しか備えておらず，使い勝手が悪いため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 TeX の編集を行う．

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，LaTeX について解説する．LaTeX とは機能強化された TeX である．LaTeX 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」という節（セクション）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，LaTeX にも様々な種類があるが，本稿で扱うのは LuaLaTeX というものである．これ以前に日本語の扱いに特化した LaTeX である pLaTeX という LaTeX があった．これをさらに改良したものが LuaLaTeX 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)

LaTeX を利用するには Windows なら，TexWorks という LaTeX 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．

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 TeX 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 TeX 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 LaTeXWorkshop の設定を変更する．冒頭に述べた通り TeX には様々な種類があるため，この設定によって LuaLaTeX がんンパイ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

[†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．
[†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．

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．このとき既になにか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ペする．そうでない人はそのままコピペすれば良い．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

```
[
  "latex-workshop.latex.outDir": "out",
  "latex-workshop.latex.autoClean.run": "onBuilt",
  "latex-workshop.latex.autoBuild.run": "onSave",
  "latex-workshop.view.pdf.viewer": "tab",
  "latex-workshop.latex.recipes": [
    {
      "name": "lualatexmk",
      "tools": ["optimized_lualatexmk"]
    }
  ],
  "latex-workshop.latex.tools": [
    {
      "name": "optimized_lualatexmk",
      "command": "latexmk",
      "args": [
        "-lualatex",
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-outdir=%OUTDIR%",
        "-auxdir=%OUTDIR%",
        "-cd",
        "-e",
        "$success_cmd = 'echo compile succeeded';",
        "%DOC%"
      ],
      "env": {
        "LUAOPT": "--luatex-cache",
        "TEXMFVAR": "%OUTDIR%/texmf-var"
      }
    }
  ]
]
```

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ペ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{newunicodecha}`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**プリアンブル**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

```
\documentclass[]{ltjsarticle}
\usepackage{graphicx} %図形の利用
\usepackage{bm} %ボールドの利用
\usepackage{newtxtext} %本文にTimes系フォントの利用
\usepackage{ascmac}%枠線で囲めるようになる
\usepackage[margin=15mm]{geometry}%余白を小さくする


%本文開始
\begin{document}
%タイトル
\title{動作確認}
\author{朕}
\date{\today}
\maketitle

朕の文章がうまく動作しておる．
\end{document}
```

4　LaTeX の書き方

4.1　LaTeX での文章の書き方

LaTeX では Word などのように書いた通りに文書は生成されない．代表的なものとして，改行や半角スペースの取り扱いがある．

LaTeX で改行を行うには，エディタ上で単に改行するだけでは無効である．改行 (段落変え) を行うには空行を挿入する必要がある．また，文章を改行するということは，パラグラフ (段落) が変わることを意味するため，改行された文章は自動的に文頭が字下げされる．

もし，事情があって単純に「改行」のみを行いたい場合は，文末に \\ を入力することで強制改行が可能である．ただし，パラグラフ内で改行してはいけない．

空白の取り扱いについても注意が必要である．挿入した半角スペースは 1 マス分のみ文書に反映され，それ以上連続して入力した半角スペースは無視される．全角スペースは文字として認識されるため，意図しない空白が入らないよう注意が必要である．

もし，スペースをあける必要がある場合には，横のスペースなら\hspace{3mm}，縦のスペースなら\vspace{3mm}などと記述することで表現可能である．

4.2　特殊文字とコメントアウト

LaTeX には，コメントアウトなどのプログラムの命令として予約されている記号がある．これらはそのまま書いても出力されず，エラーの原因となる場合がある．代表的なものに # $ % & _ { } などがある．これらを文字として出力したい場合は，直前にバックスラッシュ記号をつけ，\$ や \% のように記述する必要がある (これをエスケープ処理と呼ぶ)．

[†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 TeX のコードの pdf 化を意味する．

また，% から行末までの文字は「コメント」として扱われ，コンパイル後の文書には一切出力されない．メモ書きや，一時的に文章を隠したい場合に非常に便利である．改行は無視されるので，これを利用して例えば以下のように書くと便利である．

Listing 3　コメントアウトの利用例

```
\LaTeXとはランポート
% ランポートについて調査をする
によって作成された
% いい感じの説明を調べる
文書作成ソフトである．
```

4.3 LaTeX での図形の挿入

LaTeX で画像を挿入するには，figure 環境を使用する．Word のように画像をドラッグ&ドロップするのではなく，画像ファイルの名前を指定して読み込む方式をとる．

基本形は以下の通りである．

Listing 4　図の挿入例

```
\begin{figure}[htbp]
    \centering
    \includegraphics[width=0.8\linewidth]{fig1.png}
    \caption{図の説明}
    \label{fig:example}
\end{figure}
```

ここで重要なポイントをいくつか挙げる．

- **[htbp]**: 図の出力位置を指定するオプション．h(here), t(top), b(bottom), p(後方に page を確保) の順に優先順位を指定しており，「可能な限りその場に，無理ならページ上部に...」という意味になる．これを指定しないと図が勝手にドキュメントの最後などに飛んでいくことがある．
- **width**: 図の大きさを指定する．0.8\linewidth と指定すると，本文の幅の 80% の大きさで表示される．cm 単位などで指定することも可能である．
- **caption と label**: 図番号と参照用のラベルである．本文中で「図~\ref{fig:example}に示すように...」と参照するために必須である．ここで，チルダ (~) は改行なしの半角スペースを意味する．適切なスペースを空けるとともに，図番号のみが改行されてしまわないようにするために，図や表，参考文献の引用をする際にはこれが必要である．

4.4 LaTeX での表の作成

表は table 環境の中に tabular 環境を作成して記述する．

Listing 5　表の作成例

```
\begin{table}[htbp]
    \centering
    \caption{表の説明}
    \label{tab:example}
    \begin{tabular}{|c|l|r|} \hline
        項目 & 左寄せ & 右寄せ \\ \hline \hline
        A & Apple & 100 \\ \hline
        B & Banana & 200 \\ \hline
    \end{tabular}
\end{table}
```

`\begin{tabular}{|c|l|r|}` の部分は，列の配置を指定している．`c` は中央揃え (center)，`l` は左揃え (left)，`r` は右揃え (right) を意味し，`|` は縦線を引くことを意味する．データ同士の区切りには `&` を使用し，行の終わりには `\\` を記述する．横線を引きたい箇所には `\hline` を記述する．

見ての通り，図や表の作成は非常に記述が煩雑である．これを毎回手打ちするのは苦行に等しいため，次節で解説する「スニペット」の活用を強く推奨する．

4.5 VS Code のスニペットについて

前節までに紹介した図や表のコード，あるいは数式の定型文などを毎回手打ちしたり，過去のファイルからコピーしてくるのは非効率である．VS Code にはスニペットという機能があり，よく使うコードを登録して一瞬で呼び出すことができる．

設定方法は，VS Code の左下の歯車マークから「ユーザースニペット (User Snippets)」を選択し，`latex` を選択する（`latex.json` が開く）．ここに以下のような形式で記述する．

Listing 6　latex.json への記述例

```
"Figure Environment": {
    "prefix": "figure",
    "body": [
        "\\begin{figure}[htbp]",
        "\t\\centering",
        "\t\\includegraphics[width=${1:0.8}\\linewidth]{${2:filename}}",
        "\t\\caption{${3:caption}}",
        "\t\\label{fig:${4:label}}",
        "\\end{figure}"
    ],
    "description": "Insert a figure environment"
}
```

このように登録しておくと，LaTeX ファイルの編集画面で「figure」とタイプして Tab キーを押すだけで，上記のスニペットが展開される．`${1:0.8}` のような記述は，展開後のカーソル位置と初期値を表しており，Tab キーを押すだけで次の入力箇所（ファイル名やキャプション）へ移動できる．

よく使うプリアンブルや表組み，数式環境も同様に登録しておけば，文書作成の速度は飛躍的に向上する．「面倒な記述は一度だけ書いて保存する」のが賢い LaTeX ユーザである．

5 BibTeX による参考文献管理

参考文献の管理は大規模な文書を執筆する上で重要な問題である．仮に学位論文などを執筆する際，参考文献は一般に 30 件を超えるだろう．それらの文献情報をまとめる際，毎回書誌情報を調べ，手打ちで入力するのは大変な手間である．

この手間をほとんど 0 にしてくれるツールが BibTeX という文献管理ツールである．

5.1 BibTeX の仕組み

BibTeX では，参考文献のリストを ‘.tex’ ファイルに直接書くのではなく，‘.bib’ という拡張子の別ファイル (データベース) に保存する．

bib ファイルに記述する情報は一般に公開されている．例えば，Google Scholar などで論文や書籍を検索し，「引用」ボタンを押すと「BibTeX」というリンクが表示される．これをクリックすると以下のようなテキストが表示される．

Listing 7　references.bib の中身

```
@book{okumura2020,
  title={改訂第8版 LaTeX2e 美文書作成入門},
  author={奥村晴彦 and 黒木裕介},
  year={2020},
  publisher={技術評論社}
```

```
6}
```

これをコピーして，‘references.bib‘などの名前で保存するだけでよい．自分で著者名やタイトルを手打ちする必要はない．ここで最も重要なのは，1 行目の okumura2020 という部分である．これを**引用キー**と呼ぶ．

5.2 本文での引用方法

データベースの準備ができたら，本文中で引用キーを使って呼び出す．例えば，「奥村らによると〜」と書きたい場合，以下のように記述する．

Listing 8 本文での引用方法

```
1\documentclass{ltjsarticle}
2% （中略）
3
4\begin{document}
5
6\LaTeX の解説書として，奥村らによる書籍~\cite{okumura2020}が有名である．
7
8% 参考文献リストを出力したい場所に以下を書く
9\bibliographystyle{junsrt} % 引用順に並べるスタイル
10\bibliography{references} % .bibファイルのファイル名（拡張子不要）
11
12\end{document}
```

このように記述してコンパイルすると，\cite{...}の部分が自動的に [1] のような番号に置き換わり，巻末に参考文献リストが自動生成される．

本来，BibTeX の処理は複雑な手順 (コンパイルを 4 回繰り返すなど) が必要だが，本稿の環境構築で導入した VSCode を使用していれば，普段通り保存するだけで全自動で処理が行われる．

BibTeX では引用した参考文献のみ，参考文献一覧が生成されるため，読んだ論文の書誌情報はすべて bib ファイルの保存するようにすると良い．

6 Beamer でのスライド作成

LaTeX は論文だけでなく，プレゼンテーション用のスライドを作成することも可能である．LaTeX でスライドを作成するためのドキュメントクラスを **Beamer** と呼ぶ．

6.1 Beamer の基本構造

スライドを作成する場合，文書クラス (\documentclass) を通常の ltjsarticle から beamer に変更する必要がある．基本的なコマンドは論文作成時と同じだが，スライド 1 枚分を frame 環境で囲む点が異なる．

以下に，Beamer を使用したスライド作成の最小構成を示す．これを新しいファイル (例：slide.tex) として保存し，コンパイルすることでスライドが生成される．

Listing 9 Beamer のサンプルコード

```
1\documentclass[unicode, 12pt]{beamer}
2\usepackage{luatexja}
3
4% テーマの選択（見た目を変える）
5\usetheme{Madrid}
6
7% タイトル情報
8\title{Beamerによるスライド作成}
9\author{朕}
```

```
\date{\today}

\begin{document}

% タイトルスライド
\begin{frame}
        \titlepage
\end{frame}

% 目次スライド
\begin{frame}{目次}
        \tableofcontents
\end{frame}

% コンテンツスライド
\begin{frame}{Beamerの特徴}
        \begin{itemize}
                \item \LaTeX のコマンドがそのまま使える
                \item 数式が綺麗に表示できる
                \begin{equation}
                        e^{i\pi} + 1 = 0
                \end{equation}
                \item 朕のスライドが動いておる．
        \end{itemize}
\end{frame}

\end{document}
```

6.2 テーマの変更

Beamer の大きな特徴として，1 行書き換えるだけでデザインを一新できる「テーマ」機能がある．サンプルコード内の `\usetheme{Madrid}` の部分を，`Metropolis` や `Copenhagen` などに変更することで，好みのデザインでスライドを作成することができる．

数式を多用する理系のプレゼンテーションにおいて，LaTeX(Beamer) によるスライド作成は非常に強力なツールとなるため，ぜひ習得されたい．

参考文献

[1] 奥村，黒木,『改訂第 8 版 LaTeX 2_ε 美文書作成入門』, 技術評論社