

最強の L^AT_EX 環境構築 Windows 版

yuma, リュカ

2023 年 4 月 7 日

1 序文

曖昧さを排除したできるだけ再現性の高い L^AT_EX^{*1} 環境構築に対する処方箋を書きたいという思いでこの資料を書いた。しかし環境構築後にも難関が待ち受けている。そのために環境構築後の設定の変更等にも配慮して資料を工夫した。

- L^AT_EX を使えるようになりたい。さらに高性能エディターで書きたい。
- 自分がものすごく苦労した環境構築を 1 日であつ理解しながら終わらせたい。
- さらに、その先を自分で学習できるようにしたい。
- 苦労した環境構築の記憶をとどめておいてまた困った時に見返したい。
- 新大学 1 年生にも分かりやすく伝えたい。

という目標をもって書いた。同じような人の助けになれば幸いである。可能な限り初心者に分かりやすくそして自分が忘れないように書いたつもりである。論文は圧倒的に L^AT_EX のほうが書きやすいといわれている。この文章は、L^AT_EX を Visual Studio Code (以下 VS Code) での環境構築のために自分がしたことすべてを書いている。技術的解説もできるだけするように努めた。実は L^AT_EX の環境構築方法を完全に技術敵側面にも触れてしっかり VS Code 上に入れるための方法を体系的にまとめた資料はネットには存在しない可能性がある。もちろん分かりやすくまとめたサイトはたくさんあるが玉石混交である。そのために、VS Code 設定と、L^AT_EX 設定の両面のアプローチから環境構築設定を解説したいと思った。

1.1 編集方針

この資料は、すべて L^AT_EX によって書いた。見にくい部分もあるかもしれないがこれは、著者の技術によるところもあるので許してほしい。VS Code で L^AT_EX 文書を快適に編集する環境を構成するためには、L^AT_EX の書き方の知識、VS Code の知識、OS に関する知識、LaTeX workshop に関するさらには、.json ファイルについての知識が必要になる。こういった前提知識の多さがあるため L^AT_EX はとっつきにくいと思われることも事実である。そこで初心者にも L^AT_EX を使えるように、さらには自分好みにカスタマイズできるようになってもらうための資料とすることを編集の基本方針としたい。この資料が L^AT_EX に入門する一助となれば筆者としても幸甚の極みである。

*1 日本では「ラテフ」と読むことが多い。たまに「レイテック」「ラテック」派がいる。

1.2 この資料の特徴

この資料は、 \LaTeX の解説書ではない。VS Code で \LaTeX 文書を編集するために必要な IT 知識と設定の方法を述べる。そのあと VS Code の基本機能について説明する。美文書書作成入門 [1] は、 \LaTeX の機能の解説については大変充実した良書ではあるものの、環境構築についてはあまり触れていない。またほかのサイトなどを探しても、IT 知識が少しでもないとまったく分からない。そのためにシステムをいじりたくなくなるし、さらに取り返しのつかなくなることがまれにある。この状況を回避するためにこの資料を作った。

しかし、この資料にも至らぬ点が山ほどあると思われる。自分もすべてを理解しているわけではない。この資料の記述は自分がおこなってできたことだけを記述している。それ以外のことを知りたい場合は、付録の参考文献や LaTeX workshop の readme を読むとよい。

1.2.1 設定用コード等について

設定用コードは、コピー＆ペーストの利便性も考えて VS Code で使える markdown 形式のファイルにした。これはこの資料とともに使われることを想定したものである。

1.3 追記事項

基本編集方針にしたがってセクションの大幅改定をおこない、VS Code の機能と \LaTeX の機能、LaTeX workshop の基本機能の設定方法等についてを系統的に配置した。さらに拡張機能についてをより詳しくした。

1.4 この資料の読み方

この資料は、細かな説明などは意図せずに省かれている可能性があるため、製作者のもとでこの資料を使う形式を想定している。したがってそれ以外の場合での利用は自分で参考文献等を読まれることを強くオススメする。資料制作者の立ち合いのもとでの教授のあと使用するという形式でも十分使える資料になっている。

1.5 免責事項

この資料を使ったことによる損害等は、製作者は一切の責任を負いません。この資料に関する質問などは学術サーバー内でおこなっています。version 管理 ver2.0

目次

1	序文	1
1.1	編集方針	1
1.2	この資料の特徴	2
1.3	追記事項	2
1.4	この資料の読み方	2
1.5	免責事項	2

第 I 部 導入篇	6
2 LaTeX, VS Code とは何か？	6
2.1 LaTeX とは何か？	6
2.2 VS Code とは何か？	7
3 LaTeX に慣れる	7
3.1 cloud LaTeX の使用	7
3.2 cloud LaTeX の利用	7
第 II 部 環境構築篇	7
4 ローカル環境構築の基本用語	8
4.1 LaTeX 環境設定用 IT 用語の理解	8
4.2 LaTeX 用語の理解	9
5 LaTeXVS Code 導入実践	9
5.1 LaTeX 導入	9
5.2 Visual Studio Code 導入	9
6 環境構築実践	10
6.1 エクスプローラーの構築	10
6.2 .latexmkrc の構築	10
6.3 setting.json の設定	10
6.4 pLaTeX のコンパイルについて	11
6.5 setting.json の各種機能	11
第 III 部 LaTeX 解説篇	11
7 LaTeX の基本機能の設定	11
7.1 コンパイル	11
7.2 latexindent	12
7.3 LaTeX で書いてみる	12
8 エラーメッセージを読む	12
8.1 Recipe terminated with error	12
8.2 Recipe terminated with error. Retry building the project.	12
8.3 意味のあるエラーメッセージ	12
9 新規 package の導入	12
9.1 “sty” ファイルが入っていない場合	13
9.2 TeX Directory Structure とは何か？	13

9.3	パッケージを使うための一覧表の更新	13
9.4	使用するべき package	14
10	TeX Live の更新について	14
第 IV 部 VS Code 解説篇		14
11	VS Code の基本機能	14
11.1	、。を, . にする<置換機能>	14
11.2	コメント機能	14
11.3	各種ショートカット	15
12	VS Code の設定について	15
12.1	VS Code の設定について	15
12.2	wordwrap	15
12.3	SyncTex を使う	15
12.4	スニペット導入	16
12.5	スニペットの書き方	16
第 V 部 VS Code 拡張機能篇		17
13	LaTeX workshop の設定	18
13.1	シンタックスハイライト	18
14	VS Code の拡張機能	18
14.1	cloud LaTeX との連携	18
14.2	Ultra Math Preview の設定	19
14.3	CaTeX を導入する	19
14.4	テキスト校閲くん	19
15	キーボードショートカットの変更方法	19
16	Git 連携	19
16.1	Git の導入	19
16.2	GitHub の導入	20
第 VI 部 markdownk 解説篇		20
17	markdown とは何か?	20
18	markdown と tex ファイルの変換	21

第 I 部

導入篇

2 L^AT_EX, VS Code とは何か？

2.1 L^AT_EX とは何か？

L^AT_EX とは、文書執筆ツールとして使われる、組版処理系の一つである。L^AT_EX を使う利点をいくつか挙げる。

- 数式がきれいに書ける
 - Word で書こうとするととんでもない数のクリックと精密なエイム力を要求される^{*2}
- 見た目と論理構造を分離できる
- 修正、再利用が容易
- git でバージョン管理できる
 - 共同編集が可能
- 数学系、物理系では論文執筆のデファクトスタンダード
- 貧弱なスペックのパソコンでも編集作業がやりやすい
- 無料

VS Code の機能を使ってテンプレートを作っておけば簡単に体裁の整った文書を作成でき、内容のみに集中して執筆することが可能になる。数式交じりのレポートを作成するときなど、大学生 1 年生でも恩恵を受ける場面は多いはずである。

L^AT_EX と関わりの深い言葉として T_EX があるが、この 2 つはまったく違うものであることを注意しておこう。^{*3}たまに口頭でテフということもあるが、ユーザーが見える範囲で T_EX があらわれることはほとんどなく、大方 L^AT_EX のことを指していると考えてよい。このあたりの話は美文書書作成入門 [1] に詳しい。

この資料では、ローカル (手元の PC) に L^AT_EX をインストールし、VS Code で編集ができる環境を整える。インストール不要な環境を提供するサービスに LaTeX works editor や cloud LaTeX というものもあるが、両者を使って比較してみると、カスタマイズ性の高いローカル環境の威力がわかると思う。

T_EX の完全性

T_EX はそれ自体でチューリング完全であるから、実はほかのプログラミング言語と同様に理論上どんなことでもできる。さらに、マクロの中でもチューリング完全であることが分かっている。T_EX の可能性は無限大である。

^{*2} これは決して Word を貶しているわけではない。そのように Word を使うことが間違っているのである。Word も L^AT_EX も同じ文書執筆ツールではあるが、それぞれに長所短所があり使うべき場所というものがある。

^{*3} T_EX はプログラム言語であり、L^AT_EX は T_EX を用いて記述されたソフトウェアの集合体である。L^AT_EX は数多くのソフトウェアが組み合わさって一つの文書を出力する。

2.2 VS Code とは何か？

VS Code とは、Visual Studio Code の略でマイクロソフト社が提供する統合開発環境である。要は高性能なエディターであり、後述する拡張機能の存在によって \LaTeX との親和性も高い。VS Code は、

- 無料で頒布されている
- 動作が軽い
- クロスプラットフォームである（さまざまな OS に対応している）
- 最新トレンド全部入り
- 拡張がしやすく、さまざまな機能がある

という特徴がある。

3 \LaTeX に慣れる

美文書作成入門 [1] や一週間で \LaTeX の基礎が学べる本 [2] 等を用いて、 \LaTeX にふれることでこの後の流れが少し分かりやすくなるかもしれない。そこで、環境構築せずとも \LaTeX が使える環境として cloud LaTeX について解説する。

3.1 cloud LaTeX の使用

cloud LaTeX は、オンライン上で \LaTeX を扱うことができる。環境構築をすることなくすぐに LaTeX 文章を書くことができるという点は、 \LaTeX の記法に慣れるという面では、有用である。したがってこの章では、cloud LaTeX の使い方について簡単に説明して \LaTeX に慣れてもらいたい。

3.2 cloud LaTeX の利用

初めに <https://cloudlatex.io/ja> のサイトに行きユーザー名やメールアドレスと任意のパスワードを設定する。すると設定したメールアドレスにメールが来るので、メールアドレスの受信確認をクリックする。アカウント登録が完了する。マイページから、新規プロジェクトの追加を選んで \LaTeX 文書を書き始めることができる。cloud LaTeX は VS Code と同等の機能があるので最初に使うにはとてもいい教材である。データはすべて cloud LaTeX のサーバ上に保存されるのでデータが消える心配はほとんどない。また、共同作業も可能である。しかし、共同作業をより効率的に行いたい場合は、Overleaf のほうが優れている可能性もある。Overleaf は Git と連携することができるために論文執筆の際に有効となる。どちらがいいかは個人によるが、すぐに始めるならば cloud LaTeX をオススメする。

第 II 部

環境構築篇

4 ローカル環境構築の基本用語

4.1 L^AT_EX 環境設定用 IT 用語の理解

環境

L^AT_EX が機能するための設定やハードソフト等の存在や設定などを合わせて環境という。

ローカル

オンライン上ではなくパソコンのこと。ローカル環境とは自分のパソコンの環境のこと。

ユーザー

自分が設定した名前 自分のことユーザーの名前のフォルダーが最上位にあると考えておけばよい。

デフォルト

初期設定のこと。

隠しファイル

エクスプローラー（よくファイルを使う場所）からは見えないファイル。たとえば、.latex 等の . の前に何もついていないファイルのこと。開き方 エクスプローラーを開いて上にある表示タブを開いて表示タブにする。表示タブから隠しファイルを開いて、チェックマークをつける。これにより、今まで見えなかったファイルが見えるようになる。

ディレクトリ

ファイルが入っている階層を指し示す言葉。階層のことであり、日本語の意味では住所録の意味。ドライブ直下に置くと、最上部のフォルダーの中に入れることを意味する。ファイルの現在位置を指し示す言葉で、ルートディレクトリとは、フォルダーの階層の最上位のフォルダーを意味する。

ここで注意! L^AT_EX や、その他のプログラミング言語は日本語つまり全角のファイル名は、探すことができない。エラーを出す。このため、すべてのファイル名や、パソコンでの設定した名前は、かならず半角英数字にすることを覚えておく必要がある。

パス

ディレクトリでそのファイルの住所位置を指定するもの。

プロジェクトルート

ディレクトリの一番初めに存在するそれより上のファイルが存在しない領域

グローバル

すべてのユーザーやそのアプリケーション内ですべてに設定されるような設定のこと。

コマンドプロンプト

Windows の操作をコマンドでおこなうためのシステム。

開き方 1

コマンドプロンプトをスタートメニュー（Windows アイコンのタブ）の中の検索窓からコマンドプロンプトと打ち、マウスで出てきたコマンドプロンプトをクリックする。

開き方 2

Windows キー +R で「ファイル名を指定して実行」という窓が出てくるのでその指定する窓に

Cmd と打ち Enter を打つことで、コマンドプロンプトを実行できる。

リポジトリ

プロジェクトを構成するプログラムのソースコードやドキュメント、関連する各種のデータやファイルなどを一元的に管理する格納場所のこと

4.2 L^AT_EX 用語の理解

ビルド

L^AT_EX では、記述したソースコードに問題がないかどうかの解析を行い、問題がなければ実行可能なファイルに変換すること。

コンパイラ

コンパイラとは機械が読み取れる言語に移すことのできるソフトウェア等のこと。

コンパイル

ソースコードを機械が翻訳できる言語に移すことで、ビルドの中の一連の作業に入っている。

プリアンブル

\documentclass から \begin{document} までの中にある設定のこと。

コーディング

何らかの情報をコンピューターにコードとして書くこと。

5 L^AT_EXVS Code 導入実践

5.1 L^AT_EX 導入

L^AT_EX は、次のページからダウンロードすると良い。^{*4}

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

をクリックして、ページ上のリンク install-tl-windows.exe をクリックする。この時、警告が出るが無視して大丈夫。そして Next を押し続けて install を押す。これで、L^AT_EXpackage のほとんどすべてをダウンロードすることが可能。

インストールが正常に行われたことを確認する

インストール完了後、コマンドプロンプトを起動して、`latex -v` と打ち込む。この時に、3.141592653 のような version 情報が出てれば正常に完了している。^{*5}

5.2 Visual Studio Code 導入

Visual Studio Code は次のページからダウンロードするとよい。

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

^{*4} L^AT_EX はたくさんのソフトウェアからなるシステムであるため、配布用に必要なソフトウェア群をパッケージしたディストリビューションの形でインストールする。今回インストールするのは TeX Live というディストリビューションである。

^{*5} これは、T_EX 製作者 Donald E.Knuth 教授の意向で version up ごとに円周率 π へ近づいて行く。

をクリックしてそのあとの画面の中から今回の場合は、Windows を選択してダウンロードする。ファイルを開き、追加タスクの変更は触らないようにして Next を押し続ければ install できる。VS Code をアイコンから開いて、積み木のようなアイコン Extentions (拡張機能) をクリック。検索窓に Japanese と打って、Japanese Language Pack for Visual Studio Code を選択してインストールする。これで日本語化が完了する。これにより、VS Code 側の基本設定は終了。

6 環境構築実践

環境構築の手順を順を追って紹介する。この手法であれば今の環境下では大丈夫

6.1 エクスプローラーの構築

隠しファイルが見えるようにする。

windows 10

エクスプローラーを開いて上にある表示タブを開いて表示タブにする。表示タブから隠しファイルを開いて、チェックマークをつける。

windows 11

エクスプローラーの上部にある三点リーダーのアイコン (...) をクリックする。オプションを選択してフォルダーオプションを開く。タブから表示を選択して、「隠しファイルと隠しフォルダー、および隠しドライブを表示する」を選択する。OK を押す。

これにより、今まで見えなかったファイルが見えるようになる。

6.2 .latexmkrc の構築

latexmk とは

L^AT_EX 文書を手動でコンパイルしようとするとき複数種類のコマンドを適切な順番で複数回実行しなければならない。latexmk とは、各種 L^AT_EX のビルドコマンドを自動で一括で実行してくれるツールである。

latexmk の設定

ホームディレクトリ^{*6} 直下に .latexmkrc というテキストファイルを作成し、テキストエディタで開く。markdown ファイルに記載された設定をコピーして貼り付け、保存する。

この作業が終わったら、次は VS Code を起動する。

6.3 setting.json の設定

VS Code の設定は、setting.json^{*7} というファイルに記載されている。setting.json を開く方法はいくつかある。

^{*6} PC>ローカルディスク>ユーザー>ユーザー名、と辿れるディレクトリのこと

^{*7} .json はジェイソンと読む

1 つ目

VS Code 左下の設定マーク（歯車マーク）をクリックして、「設定」を選択する。右上端にあるファイルに矢印がついたアイコンをクリックする。

2 つ目

キーボードのショートカットキーを用いて、`ctrl+shift+P` と入力することで、コマンドパレットを出現させてそこに `Preferences:Open User Setting` と打ち込む。

`setting.json` を開いたら、`{}`の中に `markdown` ファイルに記載された設定をコピーして書き加え、保存する。もし開いた時点で何か書き込まれていた場合、`{}`の最後の要素にコンマをつけて、設定をコピーして書き加え、保存する。

6.4 p \LaTeX のコンパイルについて

p \LaTeX のコンパイルについて p \LaTeX に関する設定だけは、うまくいかないの下のように書き換える。p \LaTeX のところの `recipe tools` の変更をおこなう。`setting.json` の `latex-workshop.latex.tools` のところに次のように書き換える。と書き換えれば良い。

6.5 setting.json の各種機能

```
"latex-workshop.latex.autoBuild.run": "never",
```

は、自動ビルド機能である。しかし、自動ビルドは、このファイルのように大きすぎると非常に重くなってしまふ。 \LaTeX は、コンパイルと、書くことを分離することで、高度なレイアウトの最適化を可能にしている。これは `never` にしておくことをオススメする。

第 III 部

\LaTeX 解説篇

7 \LaTeX の基本機能の設定

\LaTeX の特徴の一つに文書の論理的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ことがある。例えば、「はじめに」という名前の節があれば文書に`\section{はじめに}`と書いておく。この文書をコンパイルするに際し、`\section{...}`という命令がどのような見た目になるかはスタイルファイルに記述されており、それを元に目的のファイルが生成される。 \LaTeX はこのような構成をしているため文章を書くときはデザインのことを考えずに済むし、デザインを変えたいときはスタイルファイルを変更するだけで済むのである。

我々が \LaTeX で文書を作る時には、この利点を最大限に享受するため、直接見た目を指示するような命令はなるべく使わないようにするべきだろう。

7.1 コンパイル

- `ctrl+Alt+B` のショートカットキーでおこなう。

- VS Code 右上の緑色の三角マークをクリックして行う。
- 左のタブの Build LaTeX project の中の Recipe:からおこなう。

ただし、今回の設定では、上の 2 つの方法でコンパイルすると Lua \LaTeX で出力される。このために、p \LaTeX などを使いたい場合は、左のタブからおこなう必要がある。

7.2 latexindent

\LaTeX のソースを自動で成形してくれるツール。これは、ソースに自動で空白等を入れてくれる環境が TeX Live にデフォルトで入っている。latexindent は VS Code 上で、shift+Alt+F をするとできる。これは itemize 環境等に空白を入れて見やすくする機能がある。

7.3 \LaTeX で書いてみる

次の \LaTeX の文章を入力する。これは、Lua \LaTeX でコンパイルするとよい。

8 エラーメッセージを読む

LaTeX workshop ではエラーがメッセージで表示されるがそれを初心者はなかなかうまく使いこなせない。そこでエラーの読み方と対応策を考える。

8.1 Recipe terminated with error

おそらく LaTeX Workshop を使ってはじめて見るだろうエラー。このエラーは、 \LaTeX 側のエラーではない。 \LaTeX のエラーは、Ctrl+Shift+M のショートカットキーで表示される。これは、注意が出ている時にも使うことができる。

8.2 Recipe terminated with error. Retry building the project.

このエラーというよりかは警告なのだがこれも本質的に意味がない。

8.3 意味のあるエラーメッセージ

意味のあるエラーメッセージは、出力される'問題'のログかまたは、“*.log”という \LaTeX のコンパイルの際に使われる中間生成物のファイルである。今回のデフォルト設定では、このファイルは削除されるため出てこないが、エラーが特定しやすくするためには、残しておいた方が良いときもある。

9 新規 package の導入

新規パッケージつまり、インターネット上や、自分で作ったスタイルファイル等を使いたいときに、めんどくさいことになる。パッケージファイルの拡張子は“.sty”であり、このパッケージを適切な場所に配置しなければならない。しかし、“.sty”が入っていない場合がある。これに対処するためには次セクション

9.1 “.sty” ファイルが入っていない場合

“.ins” ファイルが重要になる。このファイル形式は、パッケージ本体と “.ins” ファイルが入っている .dtx を L^AT_EX で .dtx ファイルを実行する必要がある。

たとえば tools.ins というファイルがあった場合には、.dtx ファイルを同じディレクトリに保存する。次にコマンドプロンプトを起動して、以下のように、入力する。

```
latex tools.ins
```

すると、同じフォルダー内に .sty ファイルが生成される。この .sty ファイルの保存場所は、次セクション参照。

9.2 TeX Directory Structure とは何か？

L^AT_EX のパッケージなどは、TDS (TeX Directory Structure) にしたがって各ディレクトリに配置されている。今回は TeX Live なので、エクスプローラーから windows フォルダーの中の texlive フォルダーの中の 2022 年フォルダーの中の texmf-dist の中の tex の中の latex フォルダーの中に各パッケージのフォルダーがあり、その中に各スタイルファイルフォルダーが保存されている。といった構成になっている。これは非常に見にくいし、読みにくいので、ここからは、これを

```
C:\texlive\2022\texmf-dist\tex\latex
```

のように書くと簡便になる。そしてこれが Windows のディレクトリの記述の仕方である。この中の tex フォルダーにはたくさんのフォルダーがあるが、最初に覚えておくべきは 2 つで、

folder 名	意味
tex	TeX の操作に関するフォルダーで、latex フォルダーがある。
font	フォントにかかわるフォルダー。

であり、このことから、.sty ファイルの保存場所は、

```
C:\texlive\2022\texmf-dist\tex\latex
```

よりも下のファイルに加えれば良いことがわかった。フォントファイルについては、以下参照。
<https://texwiki.texjp.org/?TeX%20%E3%81%AE%E3%83%87%E3%82%A3%E3%83%AC%E3%82%AF%E3%83%88%E3%83%AA%E6%A7%8B%E6%88%90>

9.3 パッケージを使うための一覧表の更新

texlive 環境では、上のように .sty ファイルを保存しても動かない。これは、ls-R というファイルが存在していてこれが、TeX が必要なパッケージを探すためのパスを与えている。そのために、この一覧表を更新しなければならない。このためには、コマンドプロンプトを開き、次のように命令するだけでいい。

```
mktexlsr
```

と入力することで一覧表を更新することができる。

9.4 使用するべき package

package は最小構成でおこなうべきである。なぜなら、Lua \LaTeX はかなり最近のものであり、package が Lua \LaTeX に対応していないということもかなりの可能性としてある。そのため、参考になるのは参考文献のサイト

10 TeX Live の更新について

第 IV 部

VS Code 解説篇

11 VS Code の基本機能

デフォルトで使える VS Code の基本機能を解説する。

11.1 、。を, . にする<置換機能>

VS Code の基本機能として、置換機能がついている。このためマクロを組まずとも、を, に変えることができる。コマンドは、Ctrl+H キーを押すと検索と置換が出てきて、Enter キーを押すとひとつずつ置換、Ctrl+Alt+Enter キーを押すとすべてが置換される。また置換を閉じるのは Esc キーである。

11.2 コメント機能

\LaTeX は % から行末までをコメントと認識する。しかしこれをいちいち書くのは面倒くさい。さらにプログラム言語によりコメント機能はまったく違う。この時に、Ctrl+/ を使うことで、その時に編集しているプログラム言語に応じて適切な構文でコメントアウトすることができる。

11.3 各種ショートカット

ショートカットキー	機能
Ctrl+Alt+B	ビルドを実行する
Ctrl+Alt+V	pdfviewer を起動する
Ctrl+click	SyncTeX の利用 (pdf 側)
Ctrl+Alt+J	SyncTeX の利用 (コード側)
Ctrl+H キー	置換パレットの表示
Ctrl+Alt+Enter	置換の全置換
Ctrl+/	行のコメントアウト機能 (全言語共通)
Ctrl+@	VS Code のターミナルの起動
Shift+Alt+F	latexindent を実行する
Ctrl+Space	コード補完を再表示
Ctrl+Shift+ + or -	ズームの程度を調整する

ただし、ショートカットキーによってビルドした場合は、Build LaTeX project の最初に記述されているコマンドをビルドコマンドとして呼び出す。これらのショートカットキーはデフォルトで設定されているもので、好きなキーに変更することもできる。

12 VS Code の設定について

12.1 VS Code の設定について

VS Code の設定は、setting.json というすべてに適用される設定と、workspace 設定という、workspace と呼ばれるファイルを何個か入れた空間について設定ができる。さらにファイルごとにも設定ができる。この優先順位は、setting.json、workspace 設定、ファイル設定の順に高い。これの設定のおかげで、さまざまな設定がおこなえるようになる。ワークスペースが 1 つのアプリケーションのように使うことができる。

12.2 wordwrap

wordwrap とは、このエディターの中で入力している中で pdfview 等に行っているときに文章が見えている間で改行を自動で行ってくれる設定である。この設定は最初、自動改行しない設定になっている。この時には、書いている行が右にシフトしてしまい、マウス又は、エンドキー等で次の行にしなければならない。また見るときも水平スクロールをしなければならない。しかしこの機能を有効にすることで、自動で改行し、文章の視認性が高くなる。したがってこの設定は、プログラミング以外では on にしておいた方がいいだろう。

12.3 SyncTex を使う

SyncTex とは何か？

TeX のソースファイルと PDF でカーソルの位置を同期する機能のこと。これを使えば、pdf 上の表示がどのソースに関係しているかということがすぐにわかる。

SyncTeX の設定

SyncTeX は、ビルドするときに形成されるファイル`*.synctex.gz`を使って動作する。今の設定では、このファイルは消去されるようになっているため、SyncTeX を使いたい場合にはこの設定を無効化する必要がある。具体的には、`setting.json` の

```
"latex-workshop.latex.clean.fileTypes":
```

の記述の中から、

```
"*.synctex.gz",
```

の記述を消去すればよい。これにより、VS Code の pdfviewer において、SyncTeX が利用できるようになる。

SyncTeX の具体的な利用の仕方

pdfviewer 上で Ctrl キーを押しながらマウスでコードをみたい場所におき、左クリックすることでそのコードの位置に飛ぶことができる。逆にコードから pdf に飛びたいときは、選択範囲をマウスで示して、またはカーソルをおいて、Ctrl+Alt+J で飛ぶことができる。(コマンドパレットから SyncTeX と入れてコマンドを実行させてもよい。)

12.4 スニペット導入

スニペットとは

スニペットとは、コードの中で何回も使うだろうコードを少しの記述でそれ呼び出す一連の動作のことをいう。スニペットの語源は、短い単語のことである。

スニペットの導入

まずスニペットを導入する。左下の歯車から「ユーザースニペットの構成」を選択する。検索窓が出てくるので、`latex` と入力して `latex.json` を開く。`latex.json` にも、`{}`があるので、その中に次のコードをコピーする。スニペットの導入は、`setting.json` とは異なるため環境を破壊することはない。

と書く。たとえば、`report` と `latex` で打つと、`report` のひな形が出てくるようになる。このように、スニペットはプリアンブル部等を簡単に早く書くことができるようになる。

12.5 スニペットの書き方

スニペットの自作フォーマットは次のように書く。スニペットも `setting.json` と同じように、`,` で各設定を区切る。

```
[caption=latex.json]
{
  "[ スニペットの名前 ]": {
    "prefix": "[ 呼び出すときのショートカット]",
    "body": [
```



```

    "[ 出力されるコードの 1 行目]",
    "[ 出力されるコードの 2 行目]",
    "...",
  ],
  "description": "[ スニペットの説明文]"
}
}

```

のように書く。

入力値の補足

入力値の中のコマンドに`$n`というのがあるがこれは、スニペットを記述するときに入力するためのカーソルが次にどこにいけば良いかを入力するための引数である。たとえば、次の例では、プログラムを載せるための環境をスニペットにより定義している。

```

[caption=latex.json]
{
  "[ スニペットの名前 ]": {
    "prefix": "[ 呼び出すときのショートカット]",
    "body": [
      "[ 出力されるコードの 1 行目 $1]",
      "[ 出力されるコードの 2 行目 $2]",
      "...",
      "$1"
      "",
    ],
    "description": "[ スニペットの説明文 ]"
  }
}

```

このように書いたときに、カーソルが`$1`を記述した後に Tab キーを押すと`$2`にカーソルが移動するように書くことができる。また、さらに、`$1`が2つあるが、これらは同じことを記述したい場合にはたとえば参照をタイトルと同じにしたい場合、両方一度に入力することができる。さらに、body の末尾に`""`を入れておけば、入力終了後に次の行からすぐ書き始めることができる。

第 V 部

VS Code 拡張機能篇

13 LaTeX workshop の設定

13.1 シンタックスハイライト

シンタックスハイライトとは、 \LaTeX の文章などを編集するためのコマンドに色をつけることでその対応やコードの可読性を高めることができる VS Code の機能である。しかしながら、LaTeX workshop はそれに対応しているのだが VS Code のカラーテーマを Visual Studio Dark または Visual Studio Light にしているとうまく機能してくれない。したがってダークテーマでシンタックスハイライトを使いたい場合は、デフォルト設定の Dark+ を使用するとよい。又はほかのカラーテーマならば良い。

14 VS Code の拡張機能

VS Code の拡張機能は、できるだけ入れないことが基本。VS Code の基本機能として使える機能があればわざわざ拡張機能を入れるべきではない。これは、VS Code の特徴である「軽い」という特徴を阻害するからである。しかし入れるべき拡張機能はもちろん存在する。

- cloud LaTeX 連携
- Ultra Math Preview
- CaTeX

14.1 cloud LaTeX との連携

cloud LaTeX を使うことでパソコン内部に \LaTeX 環境を構築しなくとも \LaTeX を使うことができる。cloud LaTeX は cloud LaTeX と同一のサーバー内とローカルのこの VS Code と連携することが可能であり、それにより、ローカルとサーバーで同期ができるために非常に便利なものとなっている。

利点

- データがサーバ上で保存するためにデータが消える心配はない。
- コンパイルがサーバ上でおこなわれるために自分のパソコンに負荷がかからないそのために自分のパソコンではできない autobuild が可能。
- cloud LaTeX のシステムやインストールされたパッケージを簡単に使うことができる。

欠点

- VS Code からサーバーを立ち上げることができないために、最初に cloud LaTeX の方に入る必要がある。
- 定期的な対応の更新をしなければならない。
- ローカル上ではコンパイルできないためにオフライン状態でコンパイルできない。

等が挙げられる。

cloud LaTeX との連携をするための設定

導入の設定は、md 形式のファイルに添付されているものを setting.json に入れればよい。cloud LaTeX 参照

14.2 Ultra Math Preview の設定

Ultra Math Preview は、LaTeX Workshop よりも強力な数式プレビューができる拡張機能である。パッケージで定義されたコマンドも Preview することができる。さらにユーザー定義の mathPreview を導入することができるために LaTeX Workshop 標準のプレビューよりもより利便性が高いものになっている。

14.3 CaTeX を導入する

CaTeX（軽鳥/怪鳥）とは、Hiromi ISHII が作った、LaTeX workshop とともに使うことを想定した、VS Code の拡張機能である。イメージ補完などが使える。これは必要になったら導入するくらいで十分であると思う。CaTeX を追加すると Ctrl+C のコマンドが使えなくなることが発生した。これは、CaTeX が勝手にショートカットキーを作るためにコピーがうまく使えなくなる。これを解決するためには、key ショートカットキーを変更しなければならない。その方法の説明はほかの章に譲る。

14.4 テキスト校閲くん

テキスト校閲くんは、tex ファイルや md ファイル等の日本語文章を校閲することができる。これはインストールするだけで利用できる。

15 キーボードショートカットの変更方法

16 Git 連携

VS Code は Git というバージョン管理システムと容易に連携させることができるようになっており、巨大な文書制作でのバージョン管理が行える。そして Git は VS Code 上のボタン操作で直感的に操作することができる。これはローカル環境にバージョン管理のためのリポジトリを生成できる。これによりバージョン管理が非常に簡単になりさらにどこを変更したのかということが非常に簡単にわかる。したがって今まで Ctrl+z で前の状態に戻っていたと思うが、セーブするとその前以上には戻ることができなかった状況からこれは理論的には無限の過去の状態に戻ることができる状態になる。なおかつローカルにもオンライン（GitHub）として共同作業をオンラインですることができたり、また全世界に公開して共同作業ができたりする。このように複数人でおこなう文書やプログラミング制作で大いに役立つ機能である。

16.1 Git の導入

git は VS Code の 3 つ目のタブにある丸が 3 つつながっているようなタブをクリックすると Git の導入を促すようなものがあるのでそれをクリックして Git のソフトウェアを Git のホームページからダウンロード

する。

16.2 GitHub の導入

GitHub は git と連携してネット上にリポジトリを生成することができる。GitHub は次のサイトから自分のメールアドレスを登録するだけで無制限に使うことができる。^{*8}<https://github.com>

第 VI 部

markdownk 解説篇

17 markdown とは何か？

この資料の付録は、markdown 記法により書いている。markdown とは、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ と同様のマークアップ言語である。この言語は html 記法と同じような性質を持つが記法をマスターするために必要な覚えるべきコマンドが非常に少ない。markdown の特色には

- 手軽に文章構造を明示できること
- 簡単で、覚えやすいこと
- 読み書きに特別なアプリを必要としないこと
- それでいて、対応アプリを使えば快適に読み書きできること

などがある。Markdown はジョン・グルーバー (John Gruber) によって 2004 年に開発され、最初は <http://daringfireball.net/projects/markdown/> で公開された。その後、多くの開発者の手を経ながら発展した。

^{*8} ついでに言うとならば Pro プランが無料である。ぜひ有効活用しよう。

18 markdown と tex ファイルの変換

付録 A ショートカットキー一覧

コマンド	機能
Alt+Z	wordwrap の変更
Ctrl+Alt+B	ビルドを実行する
Ctrl+Alt+V	pdfviewer を起動する
Ctrl+click	SyncTeX の利用 (pdf 側)
Ctrl+Alt+J	SyncTeX の利用 (コード側)
Ctrl+H キー	置換パレットの表示
Ctrl+Alt+Enter	置換の全置換
Ctrl+/	行のコメントアウト機能 (全言語共通)
Ctrl+@	VS Code のターミナルの起動
Shift+Alt+F	latexindent を実行する
Ctrl+Space	コード補完を再表示
Ctrl+Shift+M	L ^A T _E X のエラーメッセージを表示する
Ctrl+shift+M	数式環境で数式のプレビューを表示する
Ctrl+B	サイドバーの表示の設定
Ctrl+Shift++ or -	ズームの程度を調整する
Ctrl+K F	ワークスペースを閉じる

次の参考文献は、この資料を書き上げるために用いた資料一覧である。この資料を読み通すことができたならば、きっとこれらのサイトも読むことができるだろう。さらに便利な使い方や、自分なりの設定をすることもできるだろう。良い L^AT_EX ライフを

参考文献

- [1] 奥村晴彦. L^AT_EX 美文書作成入門. 技術評論社, 2020.
- [2] 明松真司. 1 週間で LaTeX の基礎が学べる本 . impress, 2022.
- [3] @passive-radio. 【大 学 生 向 け】 LaTeX 完 全 導 入 ガ イ ド Windows 編 (2022 年) . <https://qiita.com/passive-radio/items/623c9a35e86b6666b89e#4-snippet-%E3%81%AE%E3%82%B9%E3%82%B9%E3%83%A1%E6%96%87%E7%AB%A0%E4%BD%9C%E6%88%90%E3%81%AE%E3%82%82%E3%81%A3%E3%81%A8%E5%8A%B9%E7%8E%87%E5%8C%96>.
- [4] @rainbartown. VS Code で最高の LaTeX 環境を作る (2020) . <https://qiita.com/rainbartown/items/d7718f12d71e688f3573>.
- [5] 日記. 【Windows】VS Code+pLaTeX (+LuaLaTeX) 環境を構築した. <https://everykalax.hateblo.jp/entry/2022/12/15/144238>.
- [6] Web 備忘録. Web 備忘録 VS Code でスニペットを自作する方法 (2019) . <https://webbibouroku.com/Blog/Article/VSCode-snippets>.

- [7] Web 備忘録. VS Code で自作のスニペットを登録する. <https://qiita.com/michawo/items/051da6ce6d9daf9784fb>.
- [8] Web 業界で働く人を少しでも手助けするメディア. Visual Studio Code ユーザースニペットの使い方まとめ. <https://web-guided.com/620/#:~:text=%E3%82%B9%E3%83%8B%E3%83%9A%E3%83%83%E3%83%88%E3%82%92%E5%B1%95%E9%96%8B%E3%81%97%E3%81%9F%E6%99%82%E3%81%AB%E3%80%81%E3%82%AB%E3%83%BC%E3%82%BD%E3%83%AB%E3%82%92%E4%BB%BB%E6%84%8F%E3%81%AE%E4%BD%8D%E7%BD%AE%E3%81%AB%E8%A8%AD%E5%AE%9A%E3%81%99%E3%82%8B%E3%81%93%E3%81%A8%E3%81%8C%E5%87%BA%E6%9D%A5%E3%81%BE%E3%81%99%E3%80%82%20%241%20%E3%82%92%E8%A8%98%E8%BF%B0%E3%81%97%E3%81%9F%E7%AE%87%E6%89%80%E3%81%AB%E3%82%AB%E3%83%BC%E3%82%BD%E3%83%AB%E3%81%8C%E5%87%BA%E7%8F%BE%E3%81%97%E3%80%81%20%242%20%E3%80%81%20%243%20%E3%81%A8%E8%A8%98%E8%BF%B0%E3%82%92%E8%BF%BD%E5%8A%A0%E3%81%97%E3%81%9F%E5%A0%B4%E5%90%88%E3%81%AF%E3%80%81%20%EF%BC%BB,%EF%BC%BD%20%E3%82%AD%E3%83%BC%E3%82%92%E6%8A%BC%E3%81%99%E3%81%93%E3%81%A8%E3%81%A7%E3%80%81%E3%81%9D%E3%82%8C%E3%82%89%E3%81%AE%E4%BD%8D%E7%BD%AE%E3%81%AB%E3%82%AB%E3%83%BC%E3%82%BD%E3%83%AB%E3%81%8C%E7%A7%BB%E5%8B%95%E3%81%97%E3%81%BE%E3%81%99%E3%80%82%20%22body%22%3A%20%5B%20%22console.log%20%28%27%241%27%29%3B%22%2C%20%22%242%22%20%5D>.
- [9] 大橋. ルートディレクトリって結局どこ? <https://codor.co.jp/django/root-directry>.
- [10] TeX Wiki. TeX のディレクトリ構成 (TDS) . <https://texwiki.texjp.org/?TeX%20%E3%81%AE%E3%83%87%E3%82%A3%E3%83%AC%E3%82%AF%E3%83%88%E3%83%AA%E6%A7%8B%E6%88%90>.
- [11] Django Girls のチュートリアル. コマンドライン (コマンドプロンプト) とその関連について学べるサイト. <https://tutorial.djangogirls.org/ja/>.
- [12] HP 社. Windows 10 のコマンドプロンプトとは? その起動方法と使用例を紹介. <https://jp.ext.hp.com/techdevice/windows10sc/27/>.
- [13] IT 用語辞典. ビルド【build】. <https://e-words.jp/w/%E3%83%93%E3%83%AB%E3%83%89.html>.
- [14] @ta_b0_. LaTeX にソースコードを【美しく】貼る方法. https://qiita.com/ta_b0_/items/2619d5927492edbb5b03.
- [15] LuaLaTeX Lab. 【基本】 LuaLaTeX の PDF を便利にしよう～hyperref パッケージほか～. <https://lualatexlab.blog.fc2.com/blog-entry-43.html>.
- [16] KERI's Lab. VS Code で TeX を書こう. <https://www.kerislab.jp/posts/2019-01-14-VSCode-latex/#:~:text=SyncTex%20%E3%81%A8%E3%81%AF%E3%80%81TeX%20%E3%81%AE%E3%82%BD%E3%83%BC%E3%82%B9%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%81%A8%20PDF%20%E3%81%A7%E3%82%AB%E3%83%BC%E3%82%BD%E3%83%AB%E3%81%AE%E4%BD%8D%E7%BD%AE%E3%82%92%E5%90%8C%E6%9C%9F%E3%81%99%E3%82%8B%E6%A9%9F%E8%83%BD%E3%81%A7%E3%81%99%E3%80%82%20VSCode%20%E3%81%AE,%PDF%20%E3%83%93%E3%83%A5%E3%83%BC%E3%83%AF%E3%81%AF%20SyncTex%20%E3%81%AB%E5%AF%BE%E5%BF%9C%E3%81%97%E3%81%A6%E3%81%84%E3%82%8B%E3%81%AE%E3%81%A7%E3%80%81TeX%20%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%81%A8%20PDF%20%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%81%AE%E8%A9%B2%E5%BD%93%E7%AE%87%E6%89%80%E3%82%92%E8%A1%8C%E3%81%8D%E6%9D%A5%E3%81%99%E3%82%8B%E3%81%93%E3%81%A8%E3%81%8C%E3%81%A7%E3%81%8D%E3%81%BE%E3%81%99%E3%80%82>.
- [17] TeX Wiki. SyncTeX. <https://texwiki.texjp.org/?SyncTeX>.

- [18] TeX Wiki. Visual Studio Code/LaTeX. <https://texwiki.texjp.org/?Visual%20Studio%20Code%20FLaTeX#he95e080>.
- [19] TeX フォーラム. VS Code で SyncTeX が使えない. <https://okumuralab.org/tex/mod/forum/discuss.php?d=3075&parent=18242>.
- [20] @t_kemmochi. uplatex で SyncTeX するための最低限の VS Code 設定. https://qiita.com/t_kemmochi/items/dd38bbf2b823c770d1ec.
- [21] @yt1114. SyncTeX で 2 時間消費…VS Code で SyncTeX を適用するときの注意点. <https://qiita.com/yt1114/items/a50c97dafb4d193c0198>.
- [22] 情報科学屋さんを目指す人のメモ. PDF をクリックして対応する LaTeX ソースにジャンプする方法 (TeXworks+SyncTeX) . <https://did2memo.net/2015/03/05/latex-pdf-source-jump/>.
- [23] リブワークス. Visual Studio Code 完全入門. impress, 2022.
- [24] hikozaaru1202. エラーしながら学ぶ VS Code に Latex を導入 1. <https://qiita.com/hikozaaru1202/items/4189bfc52a99b7c32968>.
- [25] @hikozaaru1202. エラーしながら学ぶ VS Code に Latex を導入 2. <https://qiita.com/hikozaaru1202/items/befa7ddb6ea1b8920c92>.
- [26] @hikozaaru1202. エラーしながら学ぶ VS Code に Latex を導入 3. <https://qiita.com/hikozaaru1202/items/ce4c916f8d763b4006a9#latexmkrc%E3%81%AE%E4%BD%9C%E6%88%90>.
- [27] @Yarakashi_Kikohshi. エラーと警告を読むゾ. https://qiita.com/Yarakashi_Kikohshi/items/4ede21b06d094ad3b89e#-%E6%AD%A3%E3%81%97%E3%81%8F%E3%83%AD%E3%82%B0%E3%82%92-latex-workshop-%E3%81%AB%E5%BC%95%E3%81%8D%E6%B8%A1%E3%81%99.
- [28] @t_kemmochi. イマドキの LaTeX の書き方入門. https://qiita.com/t_kemmochi/items/78064daaa3903b7925ab#%E4%BB%98%E9%8C%B2-%E7%92%B0%E5%A2%83%E6%A7%8B%E7%AF%89.
- [29] TeX Wiki. LaTeX の警告メッセージ. <https://texwiki.texjp.org/?LaTeX%20%E3%81%AE%E8%AD%A6%E5%91%8A%E3%83%A1%E3%83%83%E3%82%BB%E3%83%BC%E3%82%B8>.
- [30] TeX Wiki. LaTeX のエラーメッセージ. <https://texwiki.texjp.org/?LaTeX%20%E3%81%AE%E3%82%A8%E3%83%A9%E3%83%BC%E3%83%A1%E3%83%83%E3%82%BB%E3%83%BC%E3%82%B8>.
- [31] san.com. CaTeX (軽鳥怪鳥) で快適 LaTeX ライフ in VS Code. <https://konn-san.com/articles/2018-11-26-happy-latex-with-catex.html>.
- [32] @skikkh. VS Code で快適 LaTeX 環境を構築する方法. <https://qiita.com/skikkh/items/707e8a5def368a69e9a6>.
- [33] TsubasaTakeda. LaTeX を VS Code で書く (初心者のレポート向け) . <https://qiita.com/TsubasaTakeda/items/bed2856c6bd427268144#%E6%8B%A1%E5%BC%B5%E6%A9%9F%E8%83%BD%E3%82%92%E8%BF%BD%E5%8A%A0>.
- [34] @TomK. VS Code のマルチカーソル練習帳. <https://qiita.com/TomK/items/3b1f5be07d708d7bd6c5>.
- [35] 12345. VS Code の便利なショートカットキー. <https://qiita.com/12345/items/64f4372fbca041e949d0>.
- [36] Yarakashi_Kikohshi. LaTeX でいろんなパッケージを usepackage する. https://qiita.com/Yarakashi_Kikohshi/items/97f9f920fb23974e0011.
- [37] Yarakashi_Kikohshi. LaTeX Workshop を使いこなす. https://qiita.com/Yarakashi_Kikohshi/

- items/a9357dd469320ffb65a0.
- [38] Yarakashi_Kikohshi. LaTeX Workshop をもう少し使いこなす. https://qiita.com/Yarakashi_Kikohshi/items/1a275f2046b002e398b3.
 - [39] Yarakashi_Kikohshi. LaTeX Workshop をもっと使いこなす. https://qiita.com/Yarakashi_Kikohshi/items/1f2225c7e28aad498998.
 - [40] Yarakashi_Kikohshi. Zotero と LaTeX Workshop で bib ファイルを扱いこなす. https://qiita.com/Yarakashi_Kikohshi/items/8f720643543ba175f7cc.
 - [41] Yarakashi_Kikohshi. LaTeX Workshop の数式プレビューを使いこなす. https://qiita.com/Yarakashi_Kikohshi/items/4570bba51787e47a03c6.
 - [42] mtk_birdman's blog. 【Windows】Visual Studio Code で Cloud LaTeX の実行環境を構築する (2022.02.16) . <https://mtkbirdman.com/windows-visual-studio-code-cloud-latex-install>.
 - [43] Yarakashi-Kikohshi. VS Code で編集して Cloud LaTeX でタイプセットする. <https://gist.github.com/Yarakashi-Kikohshi/e554045b77d35bd132eb976034625023>.
 - [44] Yarakashi-Kikohshi. なるべくデフォルトのまま使いたい人へ. https://qiita.com/Yarakashi_Kikohshi/items/e9270af54569640fe80f.
 - [45] Yarakashi-Kikohshi. Customize Syntax Highlight for LaTeX Workshop. <https://gist.github.com/Yarakashi-Kikohshi/948720cb69d0baafe71e62ec6cb2cb54>.
 - [46] James-Yu. home. <https://github.com/James-Yu/LaTeX-Workshop/wiki>.
 - [47] James-Yu. FAQ and common issues. <https://github.com/James-Yu/LaTeX-Workshop/wiki/FAQ#syntax-highlighting-does-not-work-for-most-elements>.
 - [48] かわさきしんじ. VS Code でテキストの折り返しを設定するには. <https://atmarkit.itmedia.co.jp/ait/articles/1807/27/news035.html>.
 - [49] アカリク. Cloud LaTeX Extension for Visual Studio Code. https://github.com/cloudlatex-team/cloudlatex-VSCode-extension/blob/main/docs/README_ja.md.
 - [50] migawariw. 【LaTeX】Ultra Math Preview で数式プレビューを超強化. <https://qiita.com/migawariw/items/1d3ab752f1ef261d6bcb>.
 - [51] 日本 markdown ユーザー会. <https://qiita.com/migawariw/items/1d3ab752f1ef261d6bcb>.
 - [52] 川崎庸市. Visual Studio Code の教科書. マイナビ, 2021.