

最強の L^AT_EX 環境構築 Windows 版

yuma

2023/01/09

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 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で書きたい
- 自分がものすごく苦勞した環境構築を 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。
- 苦勞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かつ自分が忘れないように書いたつもりである。L^AT_EX の世界は広すぎて分からないが word よりも数式が美しく書ける。それだけではなく、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 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 VScode 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、VScode 設定と、LaTeX 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。

この資料がどのようにほかの文書と違うのかを明確にする。この資料は、L^AT_EX の解説書ではない。L^AT_EX を VScode 上で動かすために必要な IT 知識と設定の方法を述べる。そのあと VSCode の基本機能について説明する。美文書書作成入門 [1] では VSCode 上での環境構築や何をすべきかが分からない。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全く分からない。そのためにシステムをいじりたくなくなるし、また、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。

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。VScode と LaTeX を同時に使うためには、LaTeX の書き方の知識、VSCode の知識、Windows 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。この setting.json をいじるためには、Java script の基本的な書き方の規則を知らなければならない。ゆえに初心者には環境構築は難し過ぎる。しかしそれを解決するための資料にしたいという編集の基本方針にする。

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、

追記事項

さらに便利な,SyncTeX の機能を追加するための章を加えた。VSCode の特徴と拡張機能についての章を追加した。LaTeX workshop のエラーについての章を追加した。setting.json 等のコード表を.md ファイルを用いて配布する形式に変更した。このために、配布ファイルは2つになる。

varision 管理 ver2.0

目次

1	序文	1
2	LaTeX VSCode とは何か？	3
2.1	TeX とは何か？	3
2.2	VSCode とは何か？	3
3	TeX に慣れる。	4
3.1	cloud LaTeX の使用	4
3.2	cloud LaTeX の利用	4
4	ローカル環境構築の基本用語	4
4.1	LaTeX 環境設定用 IT 用語の理解	4
4.2	LaTeX 用語の理解	5
5	LaTeX VSCode 導入篇	5
5.1	LaTeX 導入	5
5.2	Visual Studio Code 導入	5
6	環境構築篇	6
6.1	エクスプローラーの構築	6
6.2	.latexmk の構築	6
6.3	setting.json を開く	6
6.4	setting.json の設定	6
6.5	platex のコンパイルについて	7
6.6	setting.json の各種機能	7
7	LaTeX の基本機能の設定	7
7.1	コンパイル	7
7.2	Latex indent	7
7.3	LaTeX で書いてみる	7

8	エラーメッセージを読む	8
8.1	Recipe terminated with error	8
8.2	Recipe terminated with error. Retry building the project.	8
8.3	意味のあるエラーメッセージ	8
9	新規 package の導入	8
9.1	".sty" ファイルが入っていない場合	8
9.2	TeX Directory Structure とは何か?	8
9.3	パッケージを使うための一覧表の更新	9
9.4	使用するべき package	9
10	VSCoDe の基本機能	9
10.1	、。を,., にする<置換機能>	9
10.2	コメント機能	10
10.3	SyncTex を使う	10
10.4	各種コマンド	11
10.5	スニペット導入	11
10.6	スニペットの書き方	11
11	VSCoDe の拡張機能	12
11.1	L ^A T _E Xworkshop の詳細設定	13
11.2	CaTeX を導入する.	13

2 L^AT_EX VSCoDe とは何か?

2.1 T_EX とは何か?

T_EX とは, 組版ソフトウェアのことである. これを使うと数式がきれいにかけたり, 自動で組版をしてくれたりする機能がある. 詳しくは, 美文書書作成入門 [1] に詳しい. ここで, T_EX と L^AT_EX は全く違うものであることは記憶に値する. この資料では, VSCoDe と LaTeX で組版を行うが, LaTeX works editor や cloud LaTeX というものもあり, 最初はそちらを使うと VSCoDe の威力がわかると思う.

2.2 VSCoDe とは何か?

VSCoDe 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. そしてエディタとしても高性能である. VSCoDe 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. (様々な 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, 様々な機能がある.

という特徴がある.

3 TeX に慣れる.

美文書作成入門 [1] や一週間で L^AT_EX の基礎礎が学べる本 [22] 等を用いて, L^AT_EX 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 TeX が使える環境として cloud L^AT_EX について解説する.

3.1 cloud L^AT_EX の使用

cloud L^AT_EX は, オンライン上で L^AT_EX を扱うことができる. 環境構築を設定する必要なくすぐに LaTeX 文章を書くことができるために TeX 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 TeX に慣れてもらいたい.

3.2 cloud L^AT_EX 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 L^AT_EX を書き始めることができる.

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などを合わせて環境という.
- 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. 例えば, .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. 例えば, ドライブ直下に置くとは, 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! LaTeX 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. エラーが出てしまう. 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) 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 \LaTeX 用語の理解

- ビルド \LaTeX 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。
- プリアンプル `\documentclass` から `\begin{document}` までの中にある設定のこと。
- コーディング コードを各作業のこと。

5 \LaTeX VScode 導入篇

5.1 \LaTeX 導入

\LaTeX は、次のページからダウンロードすると良い。

<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、警告が出るが無視で大丈夫。そして Next を押し続けて `install` を押す。これで、 \LaTeX package のほとんどすべてをダウンロードすることが可能。

正常なインストール完了を確認する。

インストール完了後、コマンドプロンプトを起動して、`latex` と打ち込む。この時に、正式な version 情報が出てれば正常に完了している。Version は、3.141592653 のように円周率に近づいている。

(補足 これは、 \TeX 製作者 Donald E. Knuth 教授の意向で version up ごとに円周率 π に近づいて行く。)

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>

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. VSCode 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, VScode 側の基本設定は終了.

6 環境構築篇

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. 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.

6.2 .latexmk の構築

隠しファイルを開いてディレクトリをユーザーディレクトリの直下におく. このファイルはメモ帳でおこなってもよいし, VScode 上でおこなってもよい. ない場合は新たに作ること. エクスプローラーの Windows フォルダーの中のユーザーフォルダーの中に自分の名前が入っているフォルダがある. ここに次のコードを制作した名前が.latekmk という名前のファイルを保存する. つまりルートフォルダにこのファイルを保存するということ. latexmk とは, 各種 latex のビルドコマンドの実行を適切な順序で適切な回数おこなうためのツールです.

この作業が終わったら, 次は VScode を起動します.

6.3 setting.json を開く

setting.json を開くためには 2 通りある.

1 つ目

VScode 左下の設定マーク (歯車マーク) から一番上のタブのコマンドパレットから, Preferences:Open User Setting と打ちこむ方法

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, そこに Preferences:Open User Setting と打ち込む方法

の 2 つの方法があり, 熟練したら 2 番目の方法の方が効率が良い.

6.4 setting.json の設定

.json はジェイソンと呼びます. .json は Javascript に対応しています. setting.json を開いたら, そこに何か書き込まれていた場合それには触らずに, {} がかならずあるので, その中に次のコードをコピーして張り付ける.

6.5 platex のコンパイルについて

platex のコンパイルについて platex に関する設定だけは, うまくいかないので下のように書き換える. platex 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.

6.6 setting.json の各種機能

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

は, 自動コンパイル機能である. しかし, 自動コンパイルは, 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う. L^AT_EX は, コンパイルと, 書くことを分離することで, 書く必要量をへらしているために, これは never にしておくことをおすすめする.

7 L^AT_EX の基本機能の設定

L^AT_EX の基本精神は, 人間は, 内容に集中して L^AT_EX はデザインに集中するという使い方をすること. 余計な装飾はできるだけ避けるのが無難.

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.
- VSCode 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 luaL^AT_EX で出力される. このために, pLaTeX 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 Latex indent

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 TeXlive にデフォルトで入っている. latex indent は VScode 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 L^AT_EX で書いてみる.

次の L^AT_EX の文章を入力する. これは, lua L^AT_EX で書くとよい.

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 を使って初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、残しておいた方がよい可能性がある。

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 LaTeX で `.dtx` ファイルを実行する必要がある。

例えば `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？

`tex` 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 `texlive` なので、エクスプローラーから `windows` フォルダの中の `texlive` フォルダの中の 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。

```
mktextlsr
```

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、lualatex はかなり最近のものであり、package が lualatex 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 VScode の基本機能

10.1 、。を, にする<置換機能>

VScode 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づつ置換、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。

10.2 コメント機能

L^AT_EX のコメントは%である. 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全く違う. この時に, Ctrl+/ を使うことで, どんなプログラム言語にも対応したコメント機能がすぐにつけられる

10.3 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 とは, ビルドするときに形成されるファイルによって動かすことができる. つまりファイルが必要になる. そのファイルの名前は, '.synctex.gz' である. しかし, 今の setting.json では, このファイルは消去される設定になっている. ゆえに, このファイルを消去しないようにしなければならない. したがって, setting.json から

```
"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, VSCode の pdfviev 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, またはカーソルを示して, 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.)

10.4 各種コマンド

コマンド	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 (全コード共通)
Ctrl+@	VScode 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、ビルドコマンドはビルドをするのは、Build LaTeX project の最初に設定されているコマンドがデフォルトで設定されている。

10.5 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を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。例えば、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。

10.6 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。例えば、次の例では、プログラミング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、例えば参照をタイトルと同じにしたい場合に両方一度に入力することができる。また、`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に書き始めることができる。

11 VSCode の拡張機能

VSCode の拡張機能は、できるだけ入れないことが基本。VSCode 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、VSCode 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。

- LaTeX workshop 基本設定で使う。

- cloud LaTeX 連携
- CaTeX
- LaTeX Utilities

11.1 L^AT_EXworkshop の詳細設定

11.2 CaTeX を導入する.

CaTeX1(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,VSCode 上での拡張機能です. 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.

付録

ショートカットキー一覧

コマンド	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 (全コード共通)
Ctrl+@	VScode 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 ライフを送れるように

参考文献

- [1] 奥村晴彦/黒木裕介 (2020) \LaTeX 2e 美文書書作成入門
- [2] @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>
- [3] @rainbartown
VSCode で最高の \LaTeX 環境を作る (2020) <https://qiita.com/rainbartown/items/d7718f12d71e688f3573>
- [4] 日記
【Windows】 VSCode+p \LaTeX (+Lua \LaTeX) 環境を構築した
<https://everykalax.hateblo.jp/entry/2022/12/15/144238>
- [5] Web 備忘録
VSCode でスニペットを自作する方法 (2019)
<https://webbibouroku.com/Blog/Article/vscode-snippets>
- [6] @michawo
VSCode で自作のスニペットを登録する
<https://qiita.com/michawo/items/051da6ce6d9daf9784fb>
- [7] 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>
- [8] 大橋
ルートディレクトリって結局どこ？
<https://codor.co.jp/django/root-directry>
- [9] 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>
- [10] 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/>
- [11] 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/>
- [12] IT 用語辞典
 ビルド 【build】
<https://e-words.jp/w/%E3%83%93%E3%83%AB%E3%83%89.html>
- [13] @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
- [14] 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>
- [15] KERI's Lab
 VSCode 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>
- [16] TeX Wiki
 SyncTeX
<https://texwiki.texjp.org/?SyncTeX>
- [17] TeX Wiki
 Visual Studio Code/LaTeX
<https://texwiki.texjp.org/?Visual%20Studio%20Code%2FLaTeX#he95e080>
- [18] TeX フォーラム
 VScode で SyncTeX が使えない
<https://okumuralab.org/tex/mod/forum/discuss.php?d=3075&parent=18242>
- [19] @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
- [20] @yt1114
SyncTeX で 2 時間消費…VSCode で SyncTeX を適用するときの注意点
<https://qiita.com/yt1114/items/a50c97dafb4d193c0198>
- [21] 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/>
- [22] 明松真司 (2022)
1 週間で LaTeX の基礎が学べる本
- [23] リブワークス
Visual Studio Code 完全入門
- [24] @hikozaru1202
エラーしながら学ぶ VSCode に Latex を導入 1,2,3
<https://qiita.com/hikozaru1202/items/4189bfc52a99b7c32968>
<https://qiita.com/hikozaru1202/items/befa7ddb6ea1b8920c92>
<https://qiita.com/hikozaru1202/items/ce4c916f8d763b4006a9#latexmkrc%E3%81%AE%E4%BD%9C%E6%88%90>
- [25] @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
- [26] @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
- [27] 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>
- [28] 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>
- [29] san.com
CaTeX (軽鳥/怪鳥) で快適 LaTeX ライフ in VSCode
<https://konn-san.com/articles/2018-11-26-happy-latex-with-catex.html>
- [30] @skikkh
VSCode で快適 LaTeX 環境を構築する方法

- <https://qiita.com/skikkh/items/707e8a5def368a69e9a6>
- [31] TsubasaTakeda
LaTeX を VScode 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>
- [32] @TomK
VSCode のマルチカーソル練習帳
<https://qiita.com/TomK/items/3b1f5be07d708d7bd6c5>
- [33] 12345
VS Code の便利なショートカットキー
<https://qiita.com/12345/items/64f4372fbca041e949d0>
- [34] Yarakashi_Kikohshi
LaTeX でいろんなパッケージを usepackage する
https://qiita.com/Yarakashi_Kikohshi/items/97f9f920fb23974e0011
- [35] Yarakashi_Kikohshi
LaTeX Workshop を使いこなす
https://qiita.com/Yarakashi_Kikohshi/items/a9357dd469320ffb65a0
- [36] Yarakashi_Kikohshi
LaTeX Workshop をもう少し使いこなす
https://qiita.com/Yarakashi_Kikohshi/items/1a275f2046b002e398b3
- [37] Yarakashi_Kikohshi
LaTeX Workshop をもっと使いこなす
https://qiita.com/Yarakashi_Kikohshi/items/1f2225c7e28aad498998
- [38] 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
- [39] 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
- [40] 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>
- [41] Yarakashi-Kikohshi
VSCode で編集して Cloud LaTeX でタイプセットする
<https://gist.github.com/Yarakashi-Kikohshi/e554045b77d35bd132eb976034625023>