

LaTeX を始める

2022 年 6 月 18 日

目次

はじめに	2
L <small>A</small> T <small>E</small> X とは	2
プラウザで使う方法	2
Cloud LaTeX	3
Overleaf	5
ローカルで使う方法	5
TeX Live のインストール	5
VSCode 環境構築の方法	7
まとめ	17
貼ったリンク	17

はじめに

LaTeXを始める前、LaTeXを使って数式を打っているということを聞いて自分も LaTeXを打ってみたいと思ったがそもそも LaTeXとは何かということもわからず何がどうなってあの数式てんこ盛りの PDFができるんだかわからなかつたのでとりあえず LaTeXを使えるようにする方法を書く。

LaTeXとは

LaTeXとは数式を含む文書を制作して PDFにするシステムあるいはプログラミング言語である。(もともと PDFは関係ないらしいが) 基本的にはこんな画面で打つ。

The screenshot shows a Mac desktop with two windows open. On the left is a VSCodium editor window titled 'gyouretu_def.tex' containing LaTeX code. The code is a proof of L'Hopital's rule, showing the derivation of the derivative of a composition of functions. It uses TeX syntax like \begin{equation}, \lim, and \exp. Lines 139-140 show the definition of a function h'(r). Lines 141-142 show the end of the equation environment. Lines 143-144 begin a new equation environment with a split case. Lines 145-149 show the first case of the split, involving limits and derivatives. Lines 150-151 show the second case. Lines 152-153 show the end of the split and the end of the equation environment. Lines 154-155 show the start of another equation environment. Lines 156-157 show the definition of a pmatrix. Lines 158-159 show the end of the pmatrix. Lines 160-161 show the start of another equation environment. Lines 162-163 show the definition of f'(r). Lines 164-165 show the end of the equation environment. Lines 166-167 show the start of another equation environment. Lines 168-169 show the definition of e^r E. Lines 170 show the end of the equation environment. Lines 171-172 show the end of the proof section.

On the right is a LaTeX document viewer window titled 'gyouretu_def.pdf'. It displays the LaTeX code as a series of mathematical formulas and text. It includes proofs for [1] and [2], showing the derivation of the derivative of a composition of functions using the limit definition of a derivative. It also includes a diagram of a function f(r) = g(r)E + h(r)I. The proofs involve limits and derivatives of exponential and linear functions. The document viewer also shows the page number 4.

大体左側に.texのファイル、右側にpdfをおくのが主流。とはいえる普通のパソコンには当然こんなことができるアプリはデフォルトで入っていないので自分で入れるかブラウザ上で実行するかの方法で利用することになる。

ブラウザで使う方法

ブラウザから打つ方が環境構築で時間を取られない分コードに集中できるので初心者にとっては圧倒的に楽である。ブラウザで数式を含む文書を打って PDFにするのに使うツールは大体 Overleaf か Cloud LaTeX の二択だと思う。

Cloud LaTeX

Cloud LaTeXにアクセスする。



株式会社アカリクは「プライバシーマーク」付与事業者として認定されています。

メールアドレスを入力して一通り終わるところとなる。

+新規プロジェクトというところを開くと始められる。日本のものなので当然日本語入力もデフォルトで入っている。たとえ LATEX のコマンドを一切知らなくても例えばテンプレートを開いて中のテキストを打ち直す、あるいは数式をいじって数値を変えてみるといったことをしてコンパイルすれば即座に自分で数式を打ち込む快感を味わうことができる。

一番簡単なテンプレート

```
\documentclass{jsarticle}
\begin{document}
ああいい天気だ。カタカナも打てる。数式も\((a+b=c+1)\)のように打てる。
\end{document}
```

空のプロジェクトを作つて打つとこうなる。

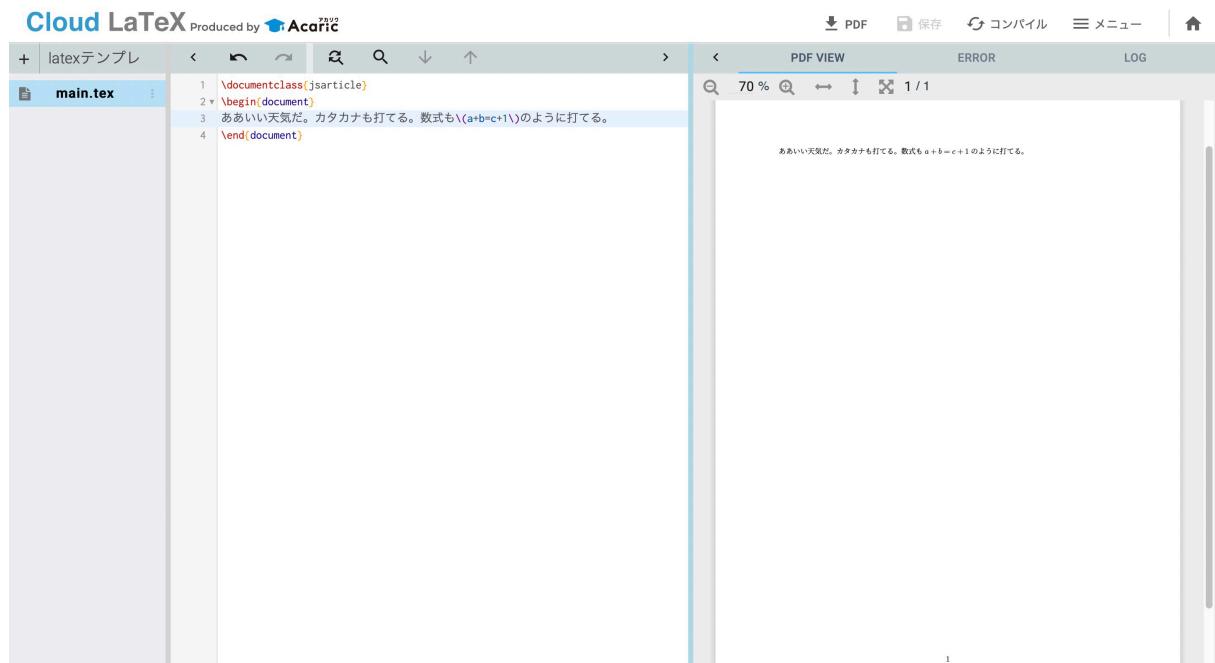


図1 4

Cloud LaTeX を使う上でターゲット設定に注意しておく必要がある。

ターゲット設定の動画

この場合は自分で作った abc.tex (名前に意味はない) をコンパイルして pdf にしようと思ってもできないことがあるが、それはターゲット設定をしていないからである。一回右クリックしてターゲット設定すればコンパイルできる。なお新しくファイルを作るには新しくプロジェクトを追加するか左上の+から「ファイルの追加」で追加すれば良い。その際 LATEX としてコンパイルしたいファイルの末尾には必ず拡張子.tex をつけることに注意する。PDF をダウンロードしたければ右上の PDF ボタンをクリックすれば良い。Cloud LaTeX は結構サジェスト機能が充実していて pandoc や platex 以外のコンパイラを使用することもできるので便利。

Overleaf

Overleafは John Hammersley と John Lees-Miller によって作られた LATEX エディタである。英語表記が標準なので、デフォルトで日本語は入っていない。よって使うには多少の設定をする必要がある。詳しくはググれば結構のってる。こちらはどちらかというと玄人向けであるが、入力中の気分の良さは上である、という人もいるかもしれない。

ローカルで使う方法

ローカルで（インターネットに繋ぐことなく）LATEX をコンパイルして pdf を作るにはそれを実行するファイルをダウンロードしてインストールして使う必要がある。そのために TeXLive、すなわち命令とインストーラが入ったものを使う。

以下、筆者が Mac ユーザーのため Mac に限って説明する。Windows の人は他の記事を見てください。あと Mac ユーザーだが MacTeX は使わない。Mac のどこに入れられるのか把握できないものが入っているからである。MacTeX 使う人は次の章まで飛ばしてください。（公式のインストーラーをポチればインストールされます。）

TeX Live のインストール

まずはTeX Live 公式サイトをブラウザで開く。

- All the ways to acquire TeX Live:
 - download

のところからダウンロードをクリックしてその先の「install-tl-unx.tar.gz」をクリックすると `install-tl-(日付)` と `install-tl-unx.tar` のファイルがダウンロードフォルダにできる。

以下ターミナルを使う。ターミナル.app は Launchpadあたりで検索すれば出てくる mac のデフォルトアプリである。ターミナルを開く。（自分は背景を黒に設定したのでちょっと違うかも知れない）

ターミナルで以下のコマンドを実行する。なお `home` は適宜自分のホームフォルダ名に変えて実行すること。

```
tar xvf "/Users/home/Downloads/install-tl-unx-2.tar"
```

によってダウンロードした texlive が展開される。`tar xvf` は展開するコマンド

```
sudo /Users/home/Downloads/install-tl-(日付)/install-tl -no-gui -repository
```

を実行すると 4000 個くらいのコンテンツが読み込まれ無事インストールされる。`sudo` は管理者権限で実行するコマンド。ターミナルから

```
platex (コンパイルしたいtexファイルの絶対パス)
```

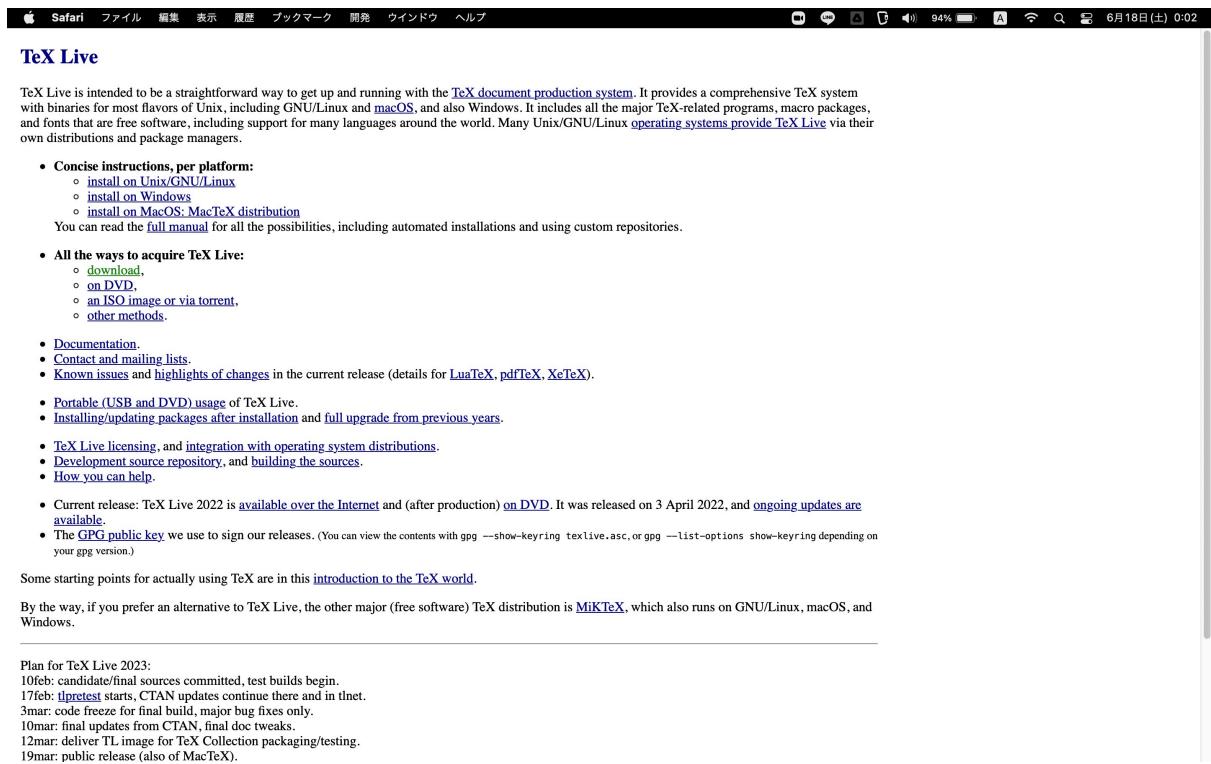


図2 5

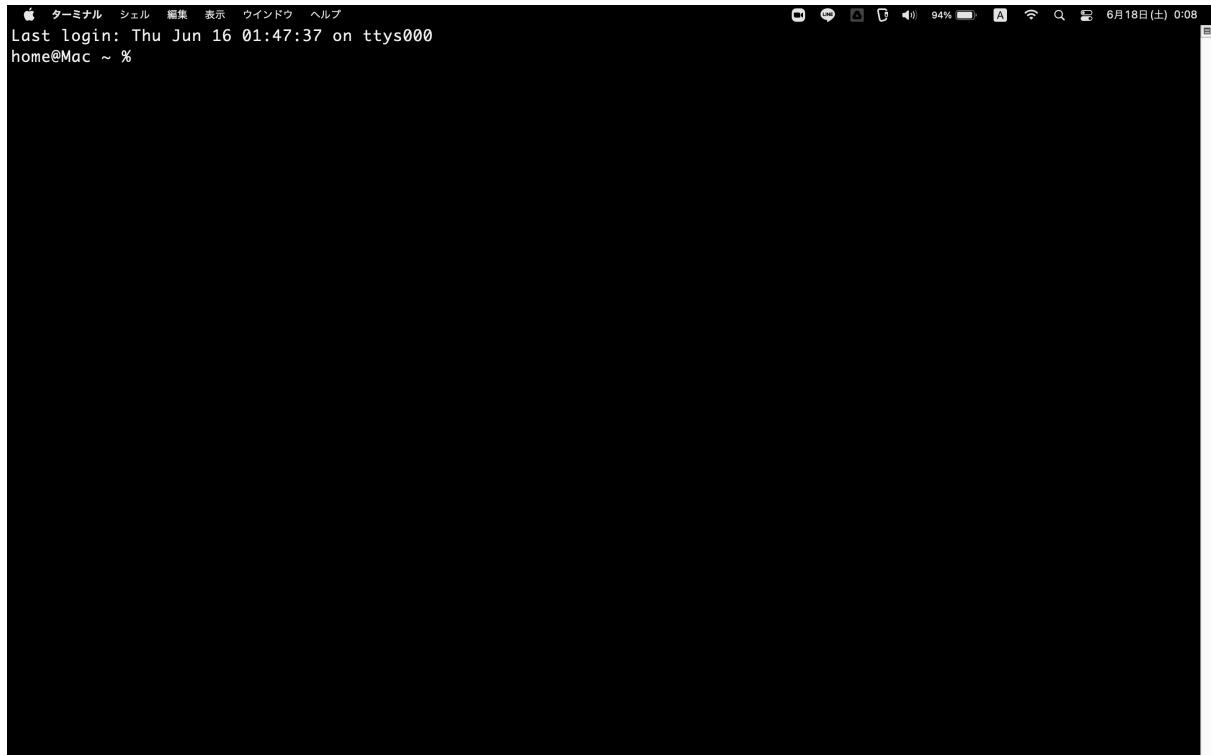


図3 6

と入力すればコンパイルできる。

さてターミナルからでも LATEX を実行できることはわかったが実際には普通に左右画面表示で tex と pdf を見比べて打てないと参考にならない。それができるもの、すなわちエディタの一つとして VSCode を取り上げる。VSCode は microsoft 社の無料エディタである。自分で設定できる予測変換機能であるスニペットが使える（一にも二にもスニペット）ので重宝する。

VSCode 環境構築の方法

macOS にかぎって説明する。VSCodeを入れる。公式サイトにアクセスする。

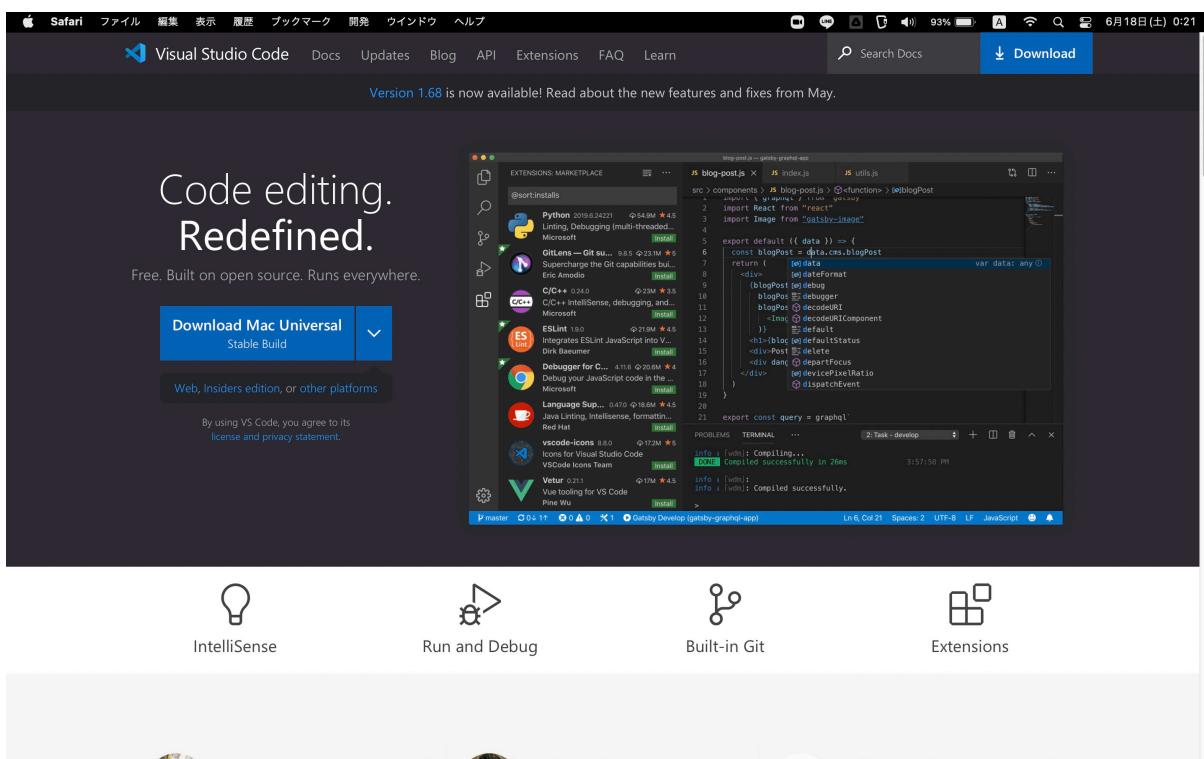


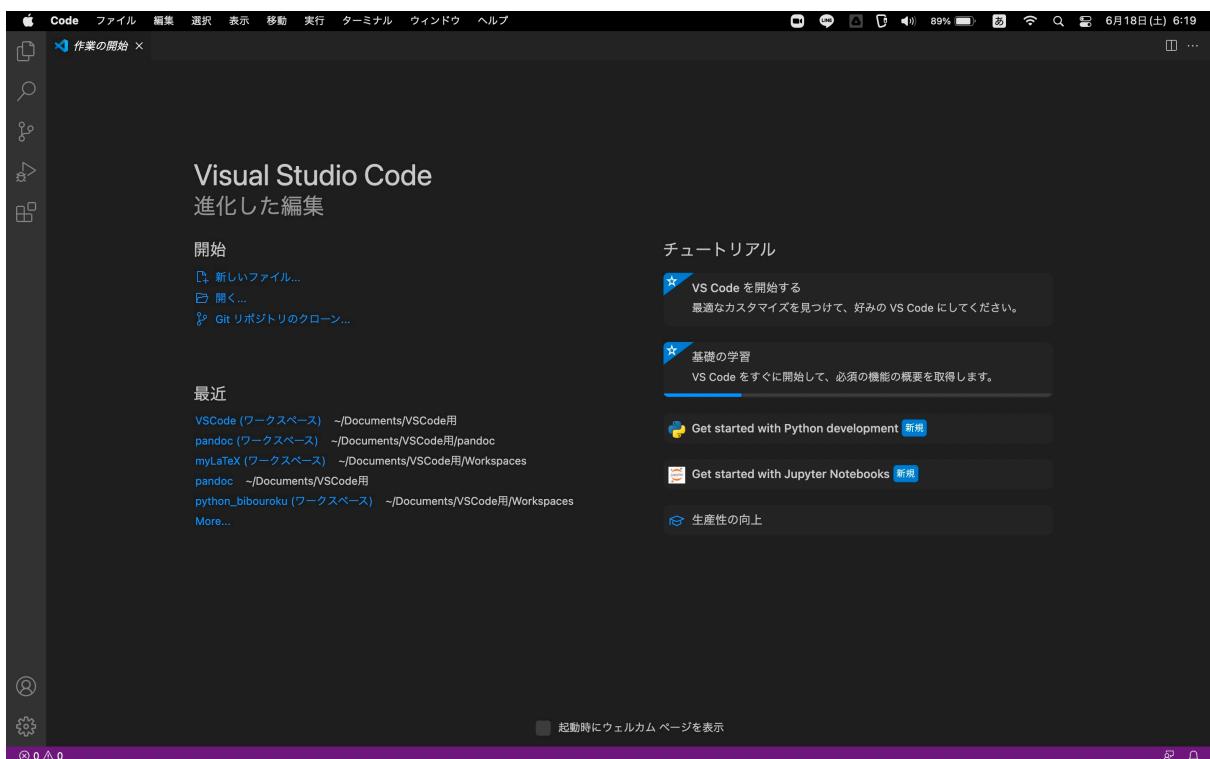
図4 7

Download Mac Universal から stable build (insiders は試作品みたいな意味っぽい) を選択してクリックする。なお VSCode は無料なので無料期間が終わったらアップグレードということはないので安心して使える。

ダウンロードしたら必ずアプリケーションフォルダに移動する。これをしないと常に/Users/home/Downloads からダウンロードする（自分はホームフォルダの名前を home にしている）から起動するというややこしいことになる。移動する際、/Applications の方に入れる。/System/Application でないことに注意。わからなければなんか後から入れたアプリがたくさん入っている方に入れれば良いと考えれば問題ない。

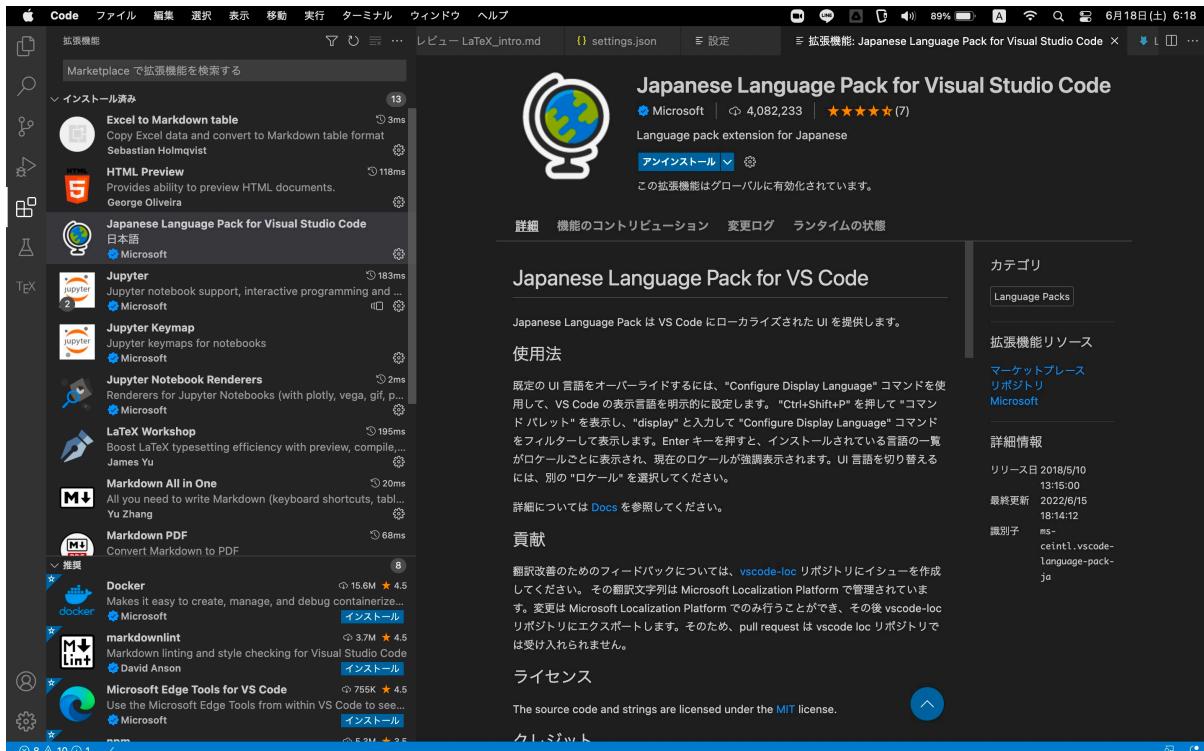


移動したら LaunchPad から Visual Studio Code を開く。

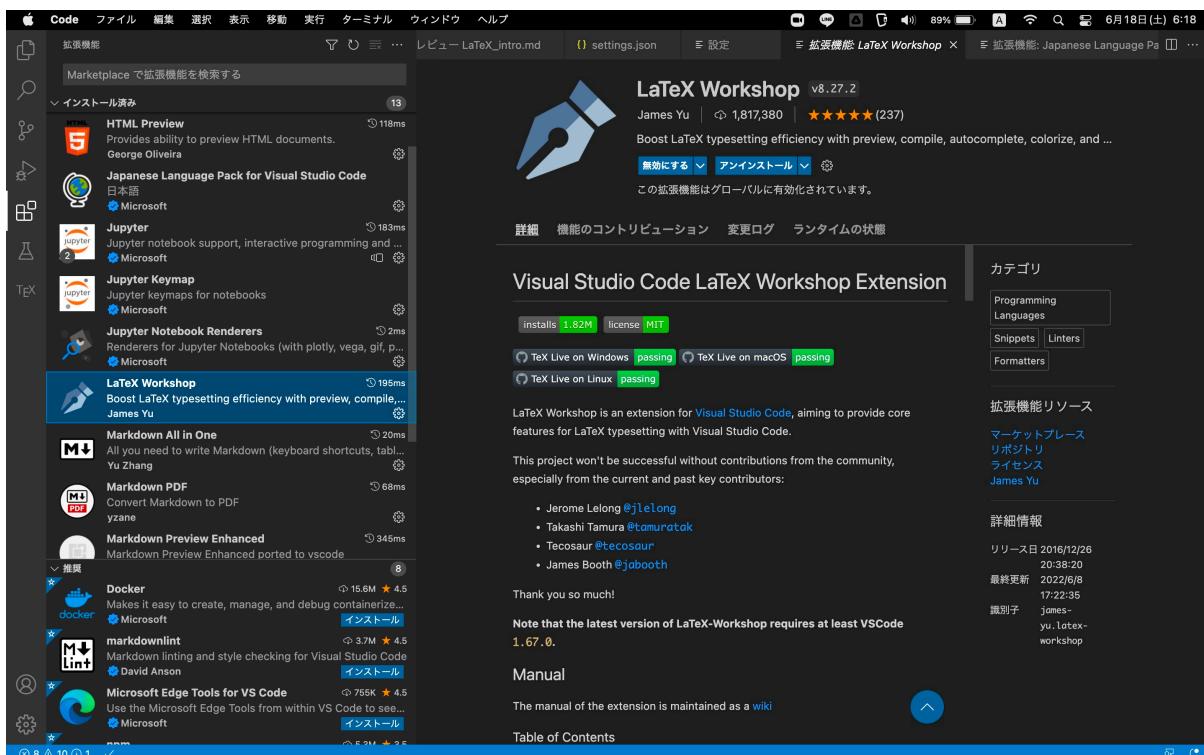


大体こんな画面が表示されるので（多分実際は英語だと思うので慌てなくて良い）まずは左の四角が四つある

ところ（拡張機能、Extension）からとりあえず Japanese Language Pack for Visual Studio Code と LaTeX Workshop をインストールする。

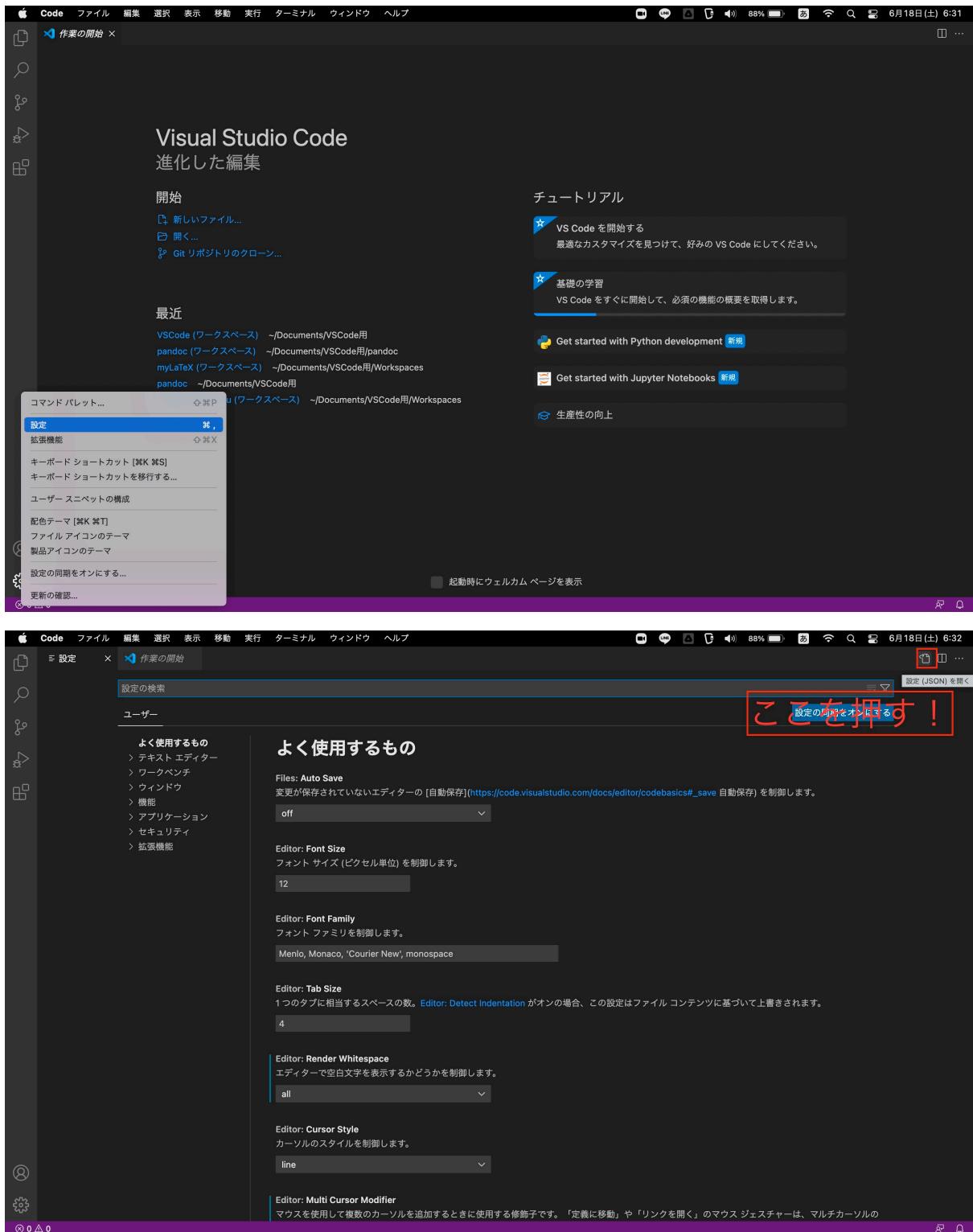


The screenshot shows the Visual Studio Code interface with the Marketplace extension open. The search bar at the top says "Marketplaceで拡張機能を検索する". The sidebar on the left lists various extensions under "インストール済み" and "推薦". The main panel displays the "Japanese Language Pack for Visual Studio Code" extension by Microsoft, which has 4,082,233 installs and a 5-star rating. The "アンインストール" button is highlighted in blue. The right sidebar shows categories like "Language Packs" and details about the extension's release date (2018/5/10) and last update (2022/6/15).



This screenshot shows the same Visual Studio Code interface with the Marketplace open again. The search bar now says "拡張機能: LaTeX Workshop". The main panel displays the "LaTeX Workshop" extension by James Yu, which has 1,817,380 installs and a 5-star rating. The "無効にする" and "アンインストール" buttons are visible. The right sidebar shows categories like "Programming Languages" and "Snippets", and details about the extension's release date (2016/12/26) and last update (2022/6/8).

なおどちらもずっと無料なのであしからず。あと下の方が青色になっているが実際はまだフォルダが入っていない状態なので紫色になる。入れたら設定を開いて setting.json を開く。



```

{
  "latex-workshop.latex.tools": [
    {
      "name": "myptools",
      "command": "latexmk",
      "args": [
        "-e", "$latex=q/platex -synctex=1 -interaction=nonstopmode -halt-on-error/",
        "-e", "$bibtex=q/pbibtex/",
        "-e", "$biber=q/biber --bblencoding=utf8 -u -U --output_safechars/",
        "-e", "$dvipdf=q/dvipdfmx %0 -o %D %S/",
        "-e", "$makeindex=q/mendex %0 -o %D %S/",
        "%DOC%",
        "-e", "$ENV{'TEXINPUTS'}='.:.' . $ENV{'TEXINPUTS'}"
      ]
    },
    {
      "name": "myxetools",
      "command": "latexmk",
      "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-xelatex",
        "-outdir=%OUTDIR%",
        "%DOC%"
      ],
      "env": {}
    },
    {
      "name": "mypdftools",
      "command": "latexmk",
      "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-pdflatex",
        "-outdir=%OUTDIR%",
        "%DOC%"
      ],
      "env": {}
    },
    {
      "name": "myluatools",
      "command": "latexmk",
      "args": [
        "-cmd",
        "-file-line-error"
      ]
    }
  ],
  "latex-workshop.latex.recipes": [
    {
      "name": "my_latex_recipes",
      "tools": [
        "myptools"
      ]
    }
  ]
}

```

setting.json は空になっているこのままでは LATEX はコンパイルできない。次の内容に書き換える。

```

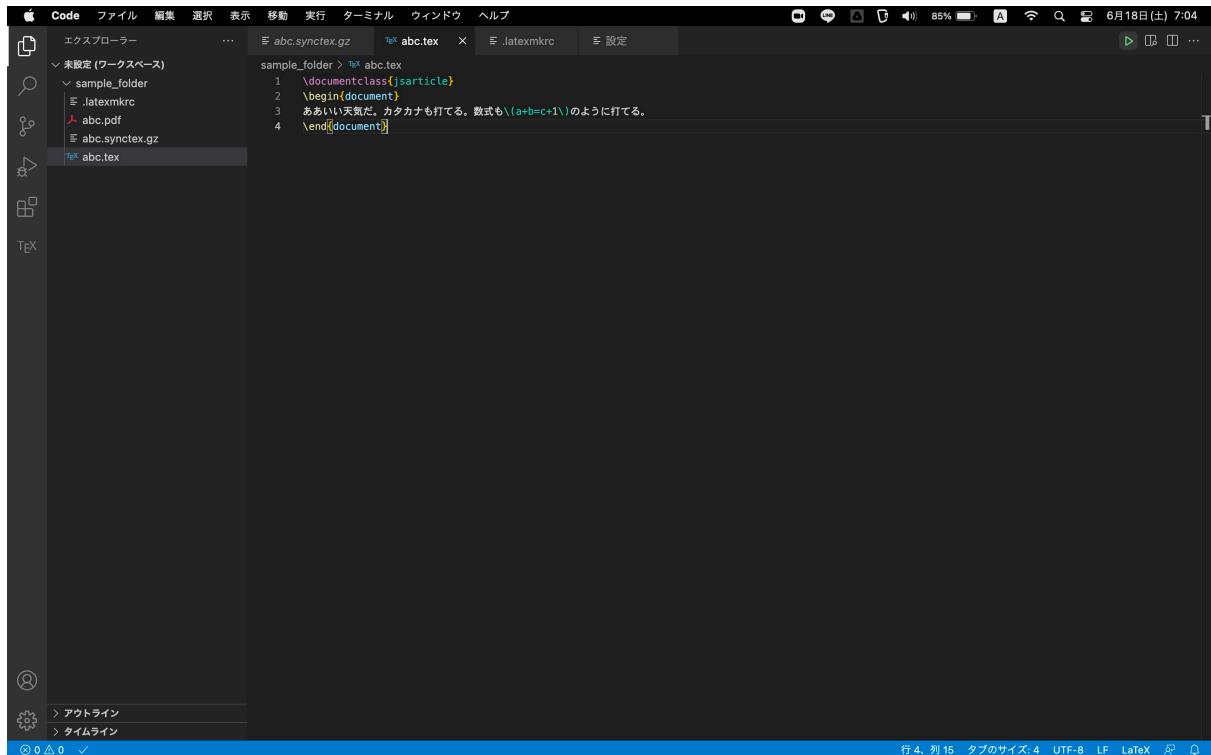
{
  "latex-workshop.latex.tools": [
    {
      "name": "myptools",
      "command": "latexmk",
      "args": [
        "-e", "$latex=q/platex -synctex=1 -interaction=nonstopmode -halt-on-error/",
        "-e", "$bibtex=q/pbibtex/",
        "-e", "$biber=q/biber --bblencoding=utf8 -u -U --output_safechars/",
        "-e", "$dvipdf=q/dvipdfmx %0 -o %D %S/",
        "-e", "$makeindex=q/mendex %0 -o %D %S/",
        "%DOC%",
        "-e", "$ENV{'TEXINPUTS'}='.:.' . $ENV{'TEXINPUTS'}"
      ]
    },
    {
      "name": "mypdftools",
      "command": "latexmk",
      "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-pdflatex",
        "-outdir=%OUTDIR%",
        "%DOC%"
      ],
      "env": {}
    },
    {
      "name": "myluatools",
      "command": "latexmk",
      "args": [
        "-cmd",
        "-file-line-error"
      ]
    }
  ],
  "latex-workshop.latex.recipes": [
    {
      "name": "my_latex_recipes",
      "tools": [
        "myptools"
      ]
    }
  ]
}

```

```
 ]
},]
}
```

`myptools` を自分で定義してそのツールをレシピとして動かす。`command` は（多分）ターミナルコマンドのことだと思う。`platex` を使ってもいいのだがこのままだと一回しかコンパイルされず目次を出力できないので勝手に何回もコンパイルしてくれるコマンド `latexmk` を使う。`bibtex` というのを見て「なんだこれは」となるが、これは目次をつけるツールであると考えて良い。デフォルトでは `latex`、`bibtex`、`dvipdf`、`makeindex` となっているところをそれぞれ `platex`、`pbtex`、`dvipdfmx`、`mendex` に書き換える。なんか p がつくと日本仕様になるくらいに考えておけば良い。

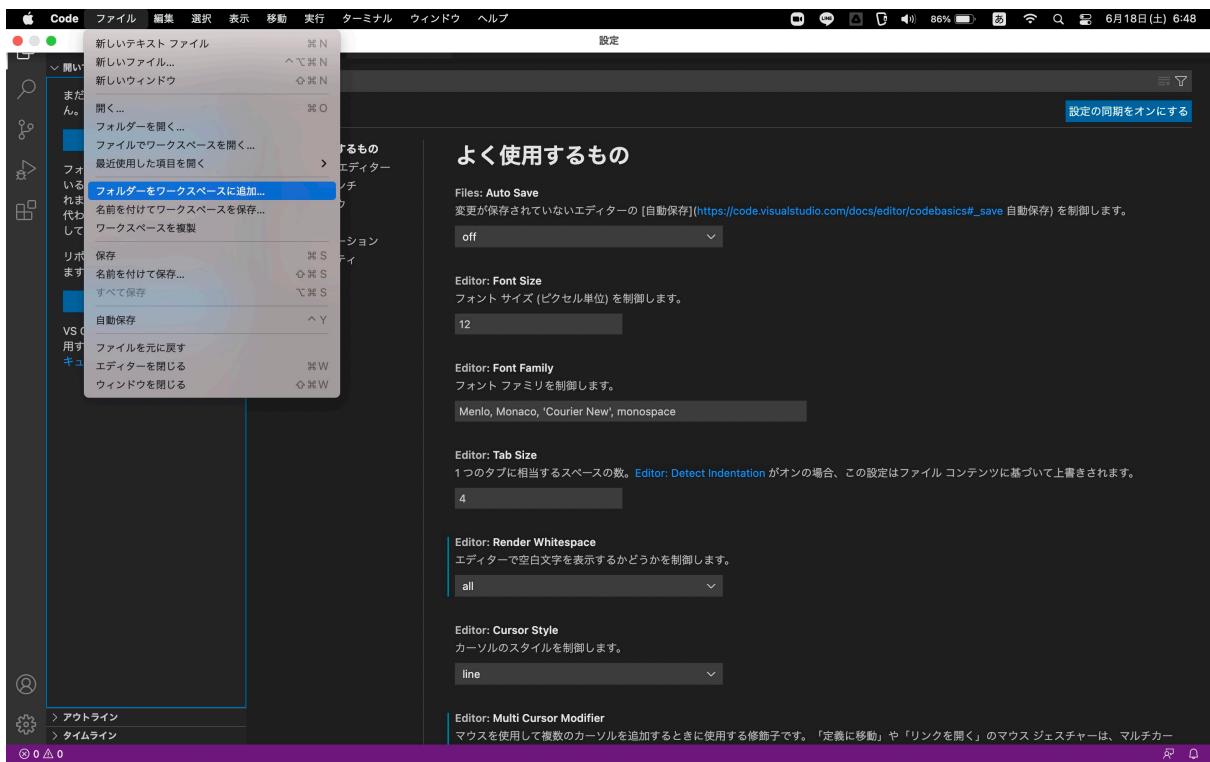
結構 `LATEX` の `setting.json` の初期設定として下のようなものが書かれているサイトがある（自分は下のような設定をしている）が後から「設定」の中で自分で英語で検索すれば出てくるのでなくてもコンパイルはできる。



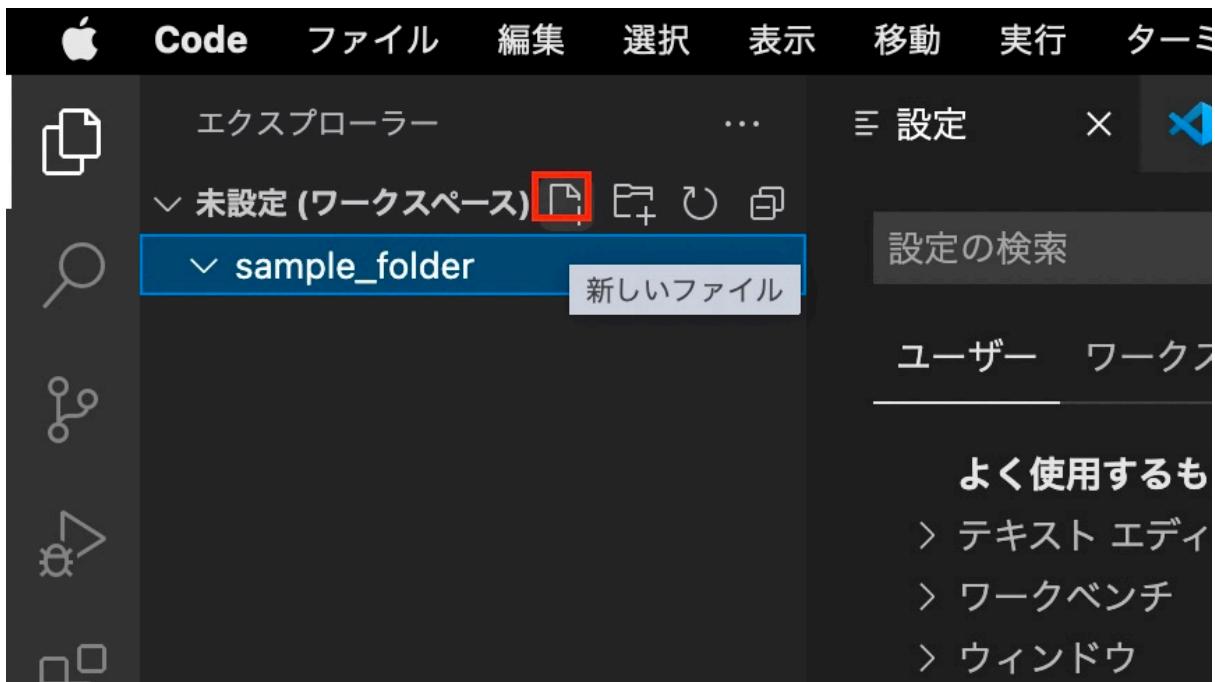
```
"latex-workshop.view.pdf.viewer": "tab",
"latex-workshop.chktex.enabled": false,
"editor.autoClosingBrackets": "beforeWhitespace",
"editor.bracketPairColorization.enabled": true,
"latex-workshop.latex.autoClean.run": "onBuilt",
"latex-workshop.latex.clean.fileTypes": [
  "*.*aux",
  "*.*bbl",
```

```
".*.blg",
".*.idx",
".*.ind",
".*.lof",
".*.lot",
".*.out",
".*.toc",
".*.acn",
".*.acr",
".*.alg",
".*.glg",
".*.glo",
".*.gls",
".*.fls",
".*.log",
".*.fdb_latexmk",
".*.snm",
".*.synctex(busy)",
".*.synctex.gz(busy)",
".*.nav",
".*.vrb",
".*.dvi",
".*.xdv"]
```

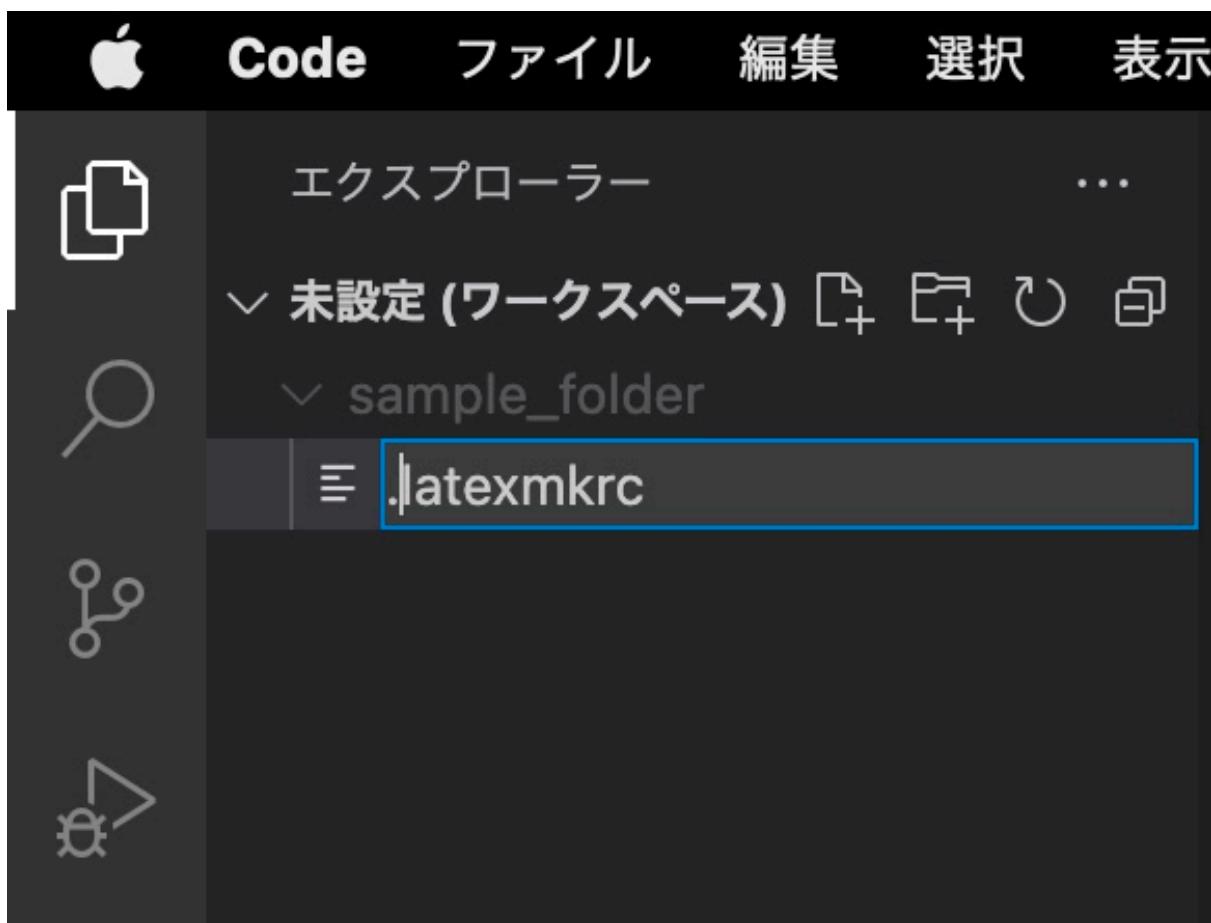
おわったら適当なところ（例えば「書類」の中）にフォルダ（名前はなんでも良い、たとえば「sample_folder」）を作る。



上のように「フォルダをワークスペースに追加...」からそれを追加する。



追加したら中に「.latexmkrc」というファイルを作成する。



中に以下を記入。

```
#!/usr/bin/env perl

# LaTeX
$latex = 'platex -synctex=1 -halt-on-error -file-line-error %0 %S';
$max_repeat = 5;

# BibTeX
$bibtex = 'pbibtex %0 %S';
$biber = 'biber --bbllencoding=utf8 -u -U --output_safechars %0 %S';

# index
$makeindex = 'mendex %0 -o %D %S';

# DVI / PDF
$dvipdf = 'dvipdfmx %0 -o %D %S';
$pdf_mode = 3;
```

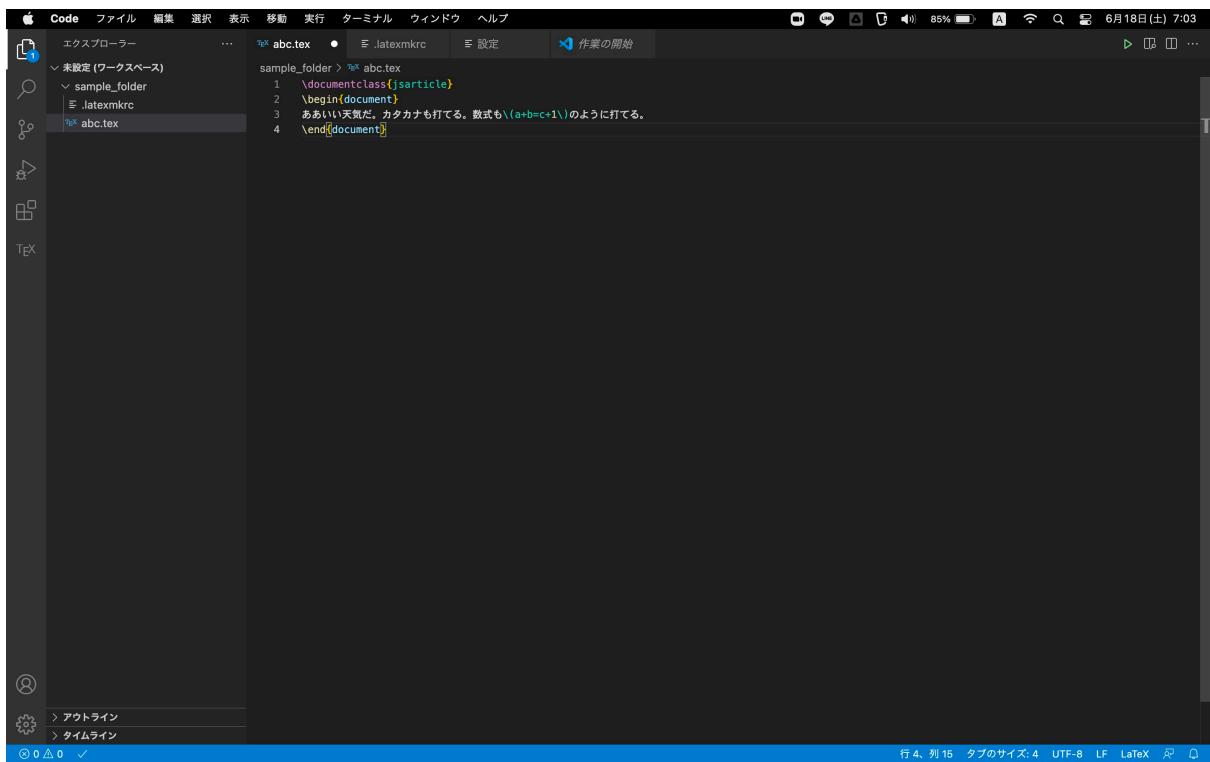
```

# preview
$pvc_view_file_via_temporary = 0;
if ($^O eq 'linux') {
    $dvi_previewer = "xdg-open %S";
    $pdf_previewer = "xdg-open %S";
} elsif ($^O eq 'darwin') {
    $dvi_previewer = "open %S";
    $pdf_previewer = "open %S";
} else {
    $dvi_previewer = "start %S";
    $pdf_previewer = "start %S";
}

# clean up
$clean_full_ext = "%R.synctex.gz"
#!/usr/bin/env perl

```

入れたら同じフォルダ（今回は sample_folder）の中に例えば「abc.tex」（名前が.texで終わればその前はなんでもいい）というファイルを作る。そこにソースコードを書いて右上の を押すとコンパイルできる。虫眼鏡みたいのを押すと pdf をプレビューできる。あるいは cmd+shift+P を押してコマンドパレットから LaTeX Workshop: Build with recipe を実行すると my_latex_recipes (先ほど setting.json で定義したレシピ名) が出てくるのでそこから一度コンパイル（ビルドともいう）すれば pdf ができる。



そんなボタン現れてないからできないよ！ていう人はもしかしたら VSCode を再起動した方がいいかも（自分はそれで解決した）

「Visual Studio Code を終了」を押すと終了するのでアプリを dock か LaunchPad から開くと再起動できる。そうすればボタンが増えてるかも知れない。

「ボタンが増えたけどコンパイルしてもなんだかわからないエラーが表示されてできない！」っていう人はもう一回再起動するか何か別の部分が違ってるのかも知れないからあんまり時間がかかるようであれば Cloud LaTeX 等プラウザ上で完結させるほうが楽な可能性もある。（スニペットは使えないものの\newcommand を駆使すれば結構なんとかなる。）

以上が VSCode の環境構築。長かった !!!

まとめ

とりあえず LATEX が打ちたい！という人のために環境構築の方法を書いた。参考になれば幸いである。

貼ったリンク

[TeX Live 公式サイト](#)

[Visual Studio Code 公式サイト](#)

Cloud LaTeX

Overleaf