

# Latex入門

## ~論文で使う文章作成ツール~

2026/4/13

スタートアップゼミ

D1 須賀

# もくじ

- Latexとは
- なぜ使うのか
- ポイント
- 使い方(VSCoDe,Overleaf)
- 研究ノート
- 文献管理

# LaTeXとは？

- 論文を書くためのツール(卒論・修論・学会論文)
- Wordとは違い「コード」で書く

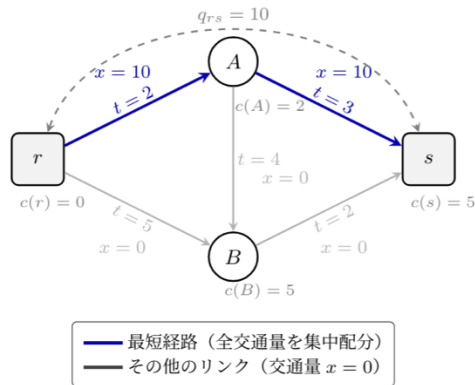


図 2.5 All-or-Nothing 配分の例 (4 ノードネットワーク, OD 交通量  $q_{rs} = 10$ ). 最短経路  $r \rightarrow A \rightarrow s$  (費用 5) に全量が配分され, 他のリンクには交通量が流れない. ノード横の  $c(i)$  は起点  $r$  からの最短経路費用を表す.

## アルゴリズム

AoN 配分は以下の手順で実施する [31, 25].

- Step 1 (初期化) 全リンクの交通量をゼロに初期化:  $x_{ij}^{(0)} = 0, n \leftarrow 1$ .
- Step 2 (最短経路計算) 起点  $o \in \mathcal{R}$  を選択する. Dijkstra 法などで起点  $o$  から全ノードへの最短経路費用  $\{c(j)\}$  および先行ポインタ  $\{F_j\}$  を計算する.
- Step 3 (ツリー順走査・集計) 最短経路ツリーを  $c(j)$  の降順 (起点から遠い順) にたどりながら, 各ノード  $j \in \mathcal{N} \setminus \{o\}$  に対し  $i := F_j$  (先行ポインタ) とおくと, リンク  $(i, j)$  の上流フロー  $y_{ij}$  を以下の式で集計する:

$$y_{ij} = q_{oj} + \sum_{m \in \mathcal{O}_j^{(o)}} y_{jm} \quad (2.5)$$

## \subsubsection\*{基本的考え方}

\textbf{All-or-Nothing (AoN) 配分}は, 各利用者がリンク旅行時間を一定と見なしたうえで最短経路のみを選択するという行動仮定に基づく \cite{土木学会1998}.

全利用者が同じ経路情報を持ち, 各ODペアの最短経路は一意に定まると仮定すると,

すべての交通量  $q_{rs}$  は1本の最短経路に集中する.

残りの経路には交通量がまったく流れないという「all or nothing (全か無か)」の割り当て方がその名前の由来である (図 \ref{fig:aon-example}).

```
\begin{figure}[hbtpt]
\centering
\input{assets/assignment/fig_aon_example.tex}
\caption{All-or-Nothing 配分の例 (4ノードネットワーク, OD交通量  $q_{rs}=10$ ).
最短経路  $r \rightarrow A \rightarrow s$  (費用5) に全量が配分され, 他のリンクには交通量が流れない.
ノード横の  $c(i)$  は起点  $r$  からの最短経路費用を表す. }
\label{fig:aon-example}
\end{figure}

\begin{table}[hbtpt]
\centering
\caption{図 \ref{fig:aon-example} のネットワークにおける最短経路費用と先行ポインタ (起点  $r$ , Dijkstra 法の出力). }
\label{tab:aon-dijkstra}
\begin{tabular}{ccc}
\toprule
ノード  $s_j$  & 最短経路費用  $c(j)$  & 先行ポインタ  $F_{j_s}$  \\
\midrule
 $r$  &  $0$  & --- \\
 $A$  &  $2$  &  $r$  \\
 $B$  &  $5$  &  $r$  \\
 $s$  &  $5$  &  $A$  \\
\bottomrule
\end{tabular}
\end{table}
```

# なぜ使うのか？

- 数式がきれい
- 図表管理が楽
- 参考文献が自動
- レイアウトが崩れない

## 参考文献

- [1] Beckmann, M., McGuire, C. B., Winsten, C. B.: Studies in the Economics of Transportation, Yale University Press, New Haven, CT, 1956.
- [2] Benders, J. F.: Partitioning procedures for solving mixed-variables programming problems, Numerische Mathematik, Vol. 4, No. 1, pp. 238–252, 1962.
- [3] Dantzig, G. B., Ramser, J. H.: The Truck Dispatching Problem, Management Science, Vol. 6, No. 1, pp. 80–91, 1959.
- [4] McFadden, D.: Conditional logit analysis of qualitative choice behavior. In Zarembka, P., ed.: Frontiers in Econometrics, Academic Press, New York, pp. 105–142, 1974.
- [5] Raissi, M., Perdikaris, P., Karniadakis, G. E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, Vol. 378, pp. 686–707, 2019.
- [6] Sandholm, W. H.: Evolutionary Implementation and Congestion Pricing, Review of Economic Studies, Vol. 69, No. 3, pp. 667–689, 2002.
- [7] Sandholm, W. H.: Population Games and Evolutionary Dynamics, MIT Press, Cambridge, MA, 2010.
- [8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention Is All You Need, Advances in Neural Information Processing Systems, Vol. 30, 2017.
- [9] Wardrop, J. G.: Some theoretical aspects of road traffic research, Proceedings of the Institution of Civil Engineers, Part II, Vol. 1, No. 3, pp. 325–362, 1952.

## 2.1.2 基本変数と関係式

交通量配分で中心的に扱う変数を表 2.1 にまとめる.

表 2.1 交通量配分の主要記号

記号	意味	単位 (例)
$x_a$	リンク $a$ の交通量	台/時
$t_a(x_a)$	リンク $a$ の旅行時間 (リンクパフォーマンス関数)	分
$t_{a0}$	リンク $a$ の自由流旅行時間 ( $x_a = 0$ のとき)	分
$C_a$	リンク $a$ の交通容量 (キャパシティ)	台/時
$f_k^{rs}$	OD ペア $rs$ の経路 $k$ の交通量	台/時
$q_{rs}$	OD ペア $rs$ の OD 交通量 (需要)	台/時
$c_k^{rs}$	OD ペア $rs$ の経路 $k$ の旅行時間	分
$u_{rs}$	OD ペア $rs$ の均衡最小旅行時間	分
$\delta_{a,k}^{rs}$	リンク $a$ が経路 $k^{rs}$ に含まれるなら 1, 含まれないなら 0	—

# ポイント

- ・最初はコピペでOK
- ・全部覚える必要なし
- ・必要なものだけ覚える

実際のコード

```
1 \documentclass{article}
2
3 \begin{document}
4
5 Hello, world!
6
7 \section{はじめに}
8
9 \[
10 E = mc^2
11 \]
12
13 \end{document}
```

基本の流れ

- ① .texファイル作成
- ② コードを書く
- ③ コンパイル (PDF生成)

# 必要なツールのインストール

## LaTeX

- Windows

：公式の `install-tl-windows.exe` をクリックしてダウンロード、GUIのインストーラーに従ってください。

- Mac

```
brew install mactex-no-gui --cask
```

- Linux (Ubuntu)

```
sudo apt install texlive-full
```

## Visual Studio Code

- Windows

公式からインストーラーをダウンロード、GUIのインストーラーに従ってください。

- Mac

```
brew install visual-studio-code --cask
```

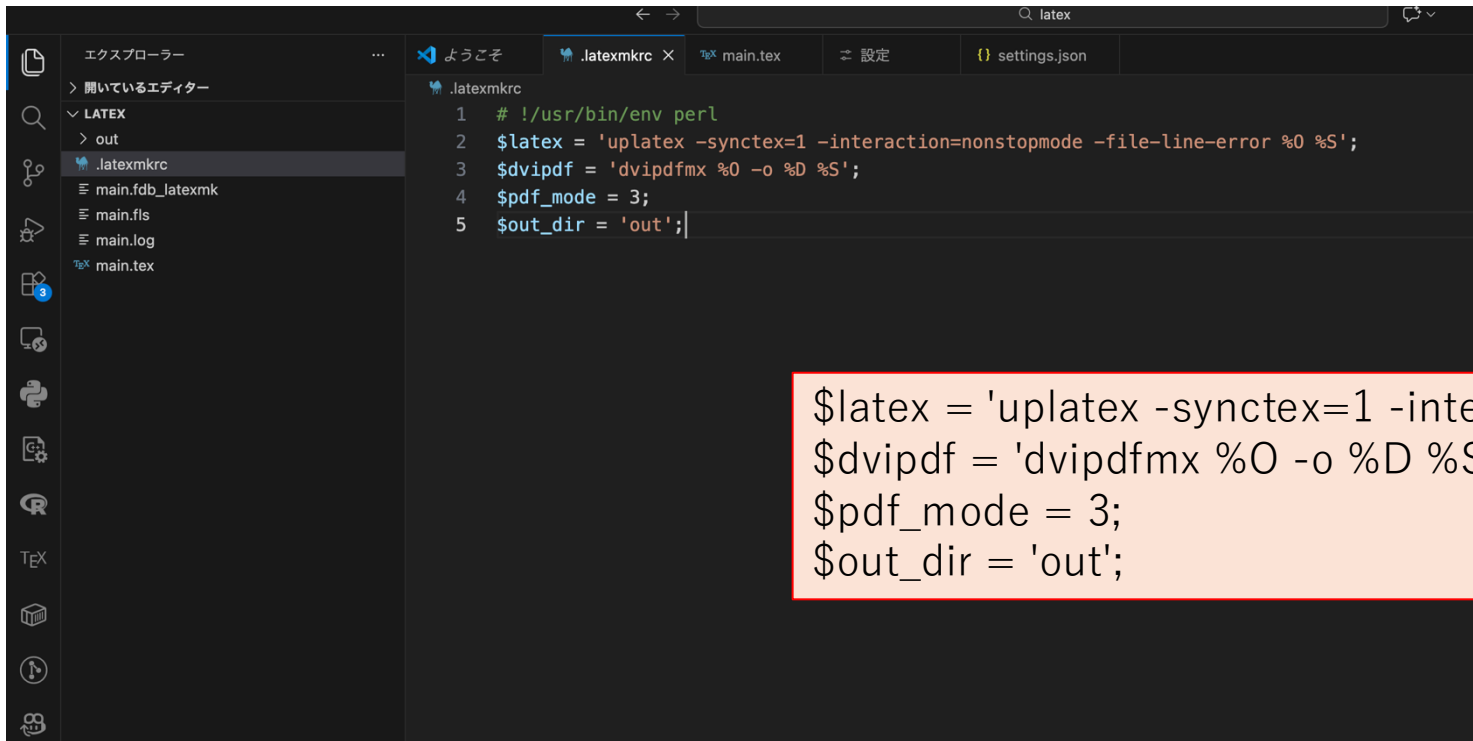
- Linux (Ubuntu)

```
sudo snap install code --classic
```

# latexmkrc の編集

「LaTeXを全部自動でコンパイルしてくれるツール」

- VSCodeを起動し、ユーザーフォルダを開きます。
- 「ファイル」-フォルダーを開く を押し、そのまま「フォルダーの選択」を押下します。
- ファイル名を **.latexmkrc** とします



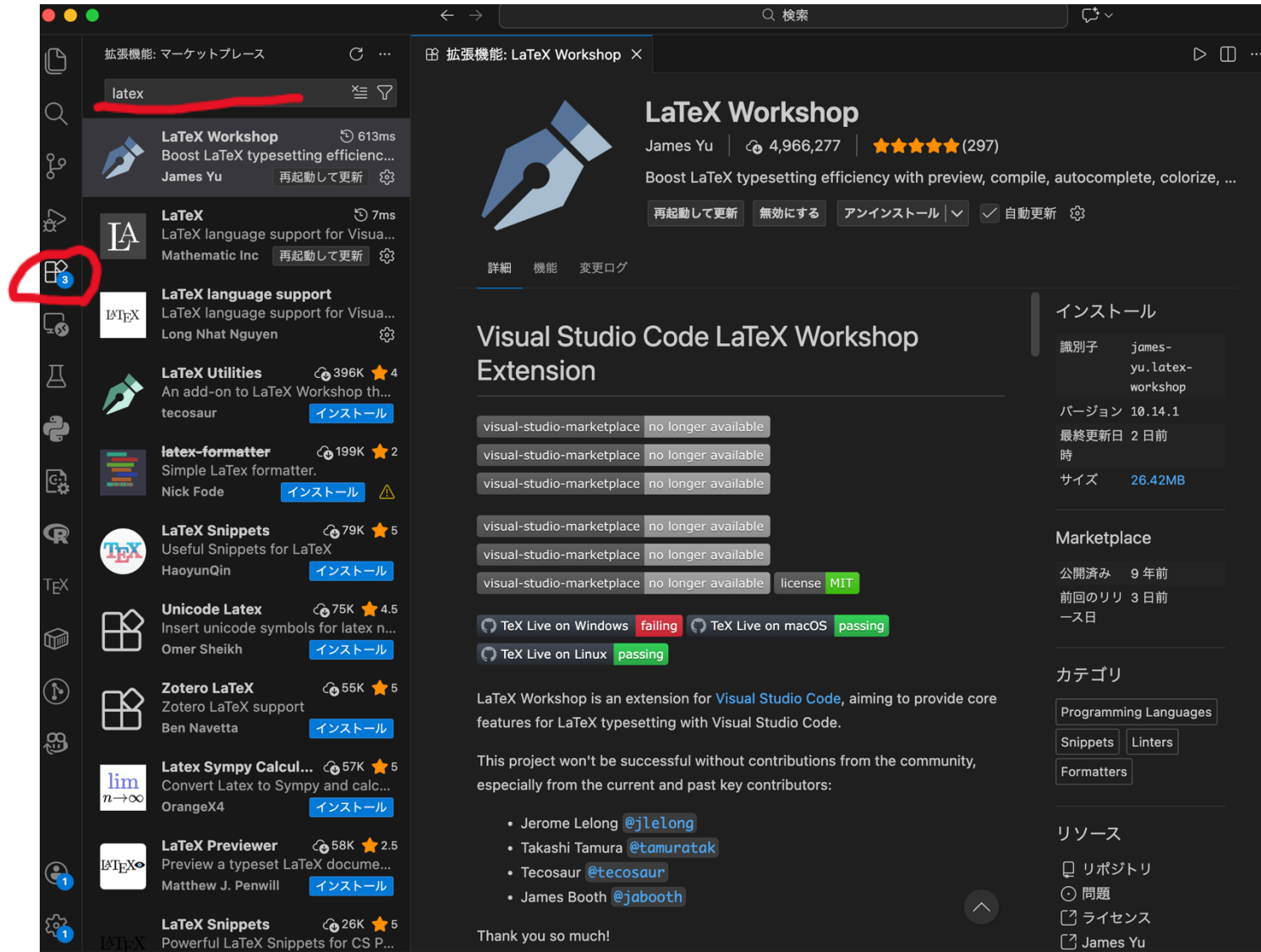
The screenshot shows the VS Code editor interface. The left sidebar displays the Explorer view with a folder named 'LATEX' containing subfolders 'out' and '.latexmkrc'. The main editor area shows the content of the '.latexmkrc' file, which is a shell script for LaTeX compilation. The code is as follows:

```
1 #!/usr/bin/env perl
2 $latex = 'uplaxex -synctex=1 -interaction=nonstopmode -file-line-error %O %S';
3 $dviPDF = 'dviPDFmx %O -o %D %S';
4 $pdf_mode = 3;
5 $out_dir = 'out';
```

```
$latex = 'uplaxex -synctex=1 -interaction=nonstopmode -file-line-error %O %S';
$dviPDF = 'dviPDFmx %O -o %D %S';
$pdf_mode = 3;
$out_dir = 'out';
```

# VSCodeでLaTeXを書く方法

## 1. 拡張機能の導入



# 拡張機能設定

The screenshot shows the Visual Studio Code interface. On the left, the Command Palette is open, listing various settings. The '設定' (Settings) option is highlighted with a red circle. In the background, the LaTeX Workshop extension page is visible, showing its details and a list of related extensions.

**Command Palette:**

- コマンドパレット...
- プロフィール
- 設定
- 拡張機能
- キーボードショートカット [⌘K ⌘S]
- スニペット
- タスク
- テーマ
- サインインして設定を同期する
- 更新プログラムのリリースノートを表示する
- 再起動して更新する (1)

**LaTeX Workshop Extension Page:**

- 拡張機能: LaTeX Workshop
- James Yu | 4,966,277 | 5 stars (297)
- Boost LaTeX typesetting efficiency with preview, compile, autocomplete, colorize, and more.
- 再起動して更新 | 無効にする | アンインストール | 自動更新
- Visual Studio Code LaTeX Workshop Extension
- visual-studio-marketplace no longer available
- TeX Live on Windows falling | TeX Live on macOS passing | TeX Live on Linux passing
- LaTeX Workshop is an extension for Visual Studio Code, aiming to provide core features for LaTeX typesetting with Visual Studio Code.
- This project won't be successful without contributions from the community, especially from the current and past key contributors:

  - Jerome Lelong @jlelong
  - Takashi Tamura @tamuratak
  - Tecosaur @tecosaur
  - James Booth @jabooth

- Thank you so much!
- Note that the latest version of LaTeX-Workshop requires at least VSCode 1.114.0 (April 2026 or later).
- Manual
- The manual of the extension is maintained as a wiki
- Table of Contents

  - Home
  - Installation and basic settings
    - Requirements
    - Installation
    - Setting PATH environment variable
    - Settings
    - Usage
  - Supported languages

拡張機能: マーケットプレース

latex

LaTeX Workshop 613ms  
Boost LaTeX typesetting efficienc...  
James Yu 再起動して更新

LaTeX 7ms  
LaTeX language support for Visua...  
Mathematic Inc 再起動して更新

LaTeX language support  
LaTeX language support for Visua...  
Long Nhat Nguyen

LaTeX Utilities 396K ★ 4  
An add-on to LaTeX Workshop th...  
tecosaur インストール

latex-formatter 199K ★ 2  
Simple LaTeX formatter.  
Nick Fode インストール

LaTeX Snippets 79K ★ 5  
Useful Snippets for LaTeX  
HaoyunQin インストール

Unicode Latex 75K ★ 4.5  
Insert unicode symbols for latex n...  
Omer Sheikh インストール

Zotero LaTeX 55K ★ 5  
Zotero LaTeX support  
Ben Navetta インストール

Latex Sympy Calcul... 57K ★ 5  
Convert Latex to Sympy and calc...  
OrangeX4 インストール

LaTeX Previewer 58K ★ 2.5  
Preview a typeset LaTeX docume...  
Matthew J. Penwill インストール

LaTeX Snippets 26K ★ 5  
Powerful LaTeX Snippets for CS P...  
sabertazimi インストール

LaTeX Songs 11K ★ 5  
Syntax highlighting for LaTeX son...  
webartoli インストール

Cloud LaTeX 20K  
Compile LaTeX document with Cl...  
cloudlatex インストール

LaTeX Extension Pack 19K ★ 5  
Collection of extensions related t...

拡張機能: LaTeX Workshop

設定

設定の検索

最終同期: 2 か月前

よく使用するもの

ユーザー

よく使用するもの

Editor: Font Size  
フォント サイズ (ピクセル単位) を制御します。  
16

Editor: Format On Save  
 保存時にファイルを書式設定します。フォーマッタが使用可能である必要があり、エディターをシャットダウンすることはできません。Files: Auto Save が 'afterDelay' に設定されている場合、ファイルは明示的に保存されたときにのみ書式設定されます。

Files: Auto Save  
変更が保存されていないエディターの 自動保存 を制御します。  
afterDelay

Editor: Default Formatter  
他のすべてのフォーマッタ設定よりも優先される、既定のフォーマッタを定義します。フォーマッタを提供している拡張機能の識別子にする必要があります。  
なし

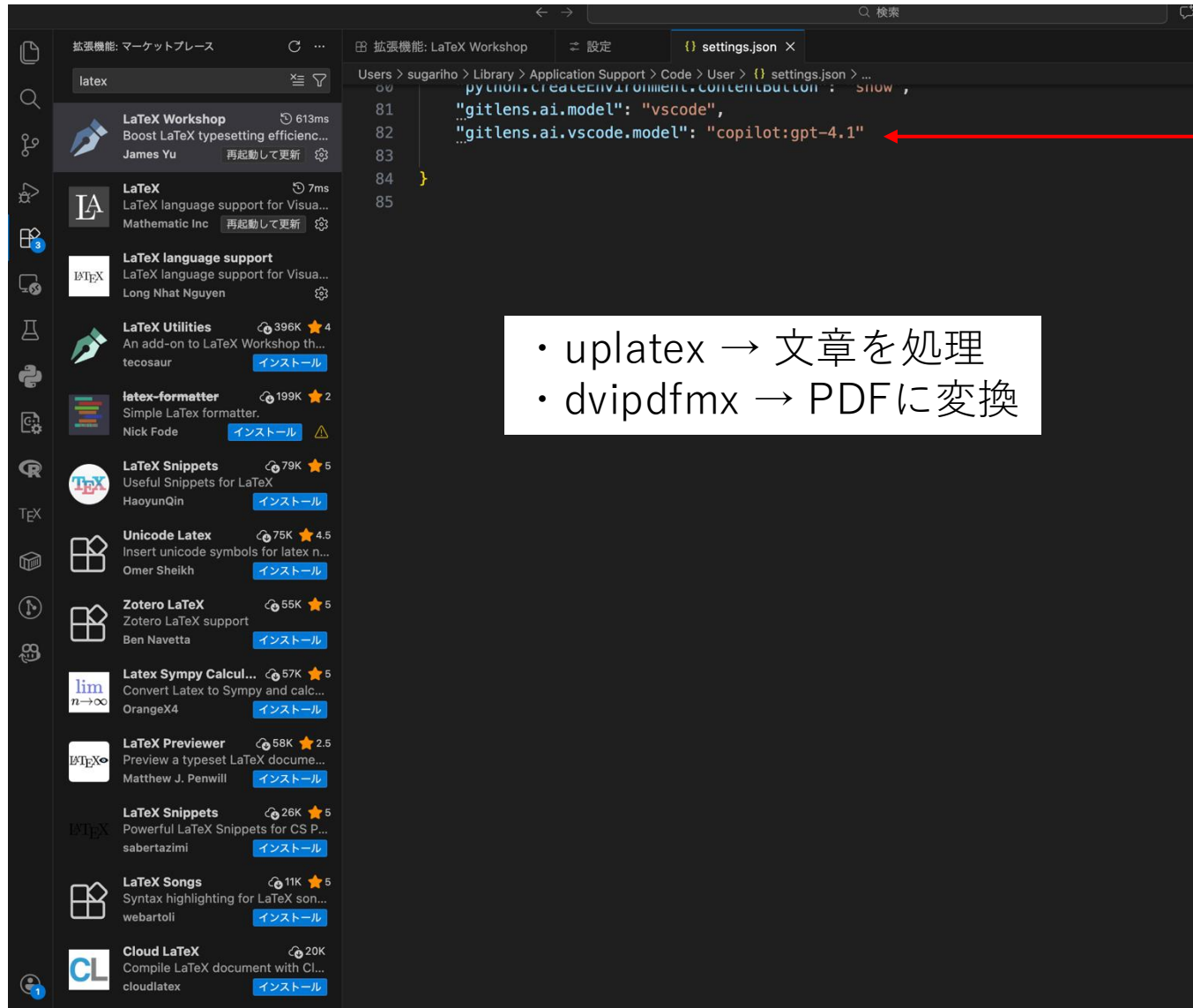
Editor: Font Family  
フォント ファミリを制御します。  
Menlo, Monaco, 'Courier New', monospace

Editor: Word Wrap (他の場所に変更済み)  
行の折り返し方法を制御します。  
off

Files: Exclude  
ファイルとフォルダーを除外するための glob パターンを構成します。たとえば、エクスプローラーでは、この設定に基づいて表示または非表示にするファイルとフォルダーが決定されます。検索固有の除外を定義するには、Search: Exclude 設定を参照してください。|.gitignore に基づいてファイルは無視するには、Explorer: Exclude Git Ignore 設定を参照してください。  
\*\*/.git  
\*\*/.svn  
\*\*/.hg  
\*\*/.DS\_Store  
\*\*/Thumbs.db  
パターンを追加

# settings.json の編集

「どうやってコンパイルするか」などをここで決めます。



- uplatex → 文章を処理
- dvipdfmx → PDFに変換

すでに何が書かれていたら、最後の項目の末尾に、を追加してから以下を貼り付ける

```
{
  "latex-workshop.latex.autoBuild.run": "onSave",

  "latex-workshop.latex.tools": [
    {
      "name": "uplatex",
      "command": "uplatex",
      "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "%DOC%"
      ]
    },
    {
      "name": "dvipdfmx",
      "command": "dvipdfmx",
      "args": [
        "%DOCFILE%.dvi"
      ]
    }
  ],

  "latex-workshop.latex.recipes": [
    {
      "name": "uplatex -> dvipdfmx",
      "tools": [
        "uplatex",
        "dvipdfmx"
      ]
    }
  ]
}
```

# 実際に使ってみよう



```
\documentclass[uplatex]{jsarticle}

\usepackage[dvipdfmx]{graphicx}

\begin{document}

こんにちは！

\section{テスト}

これは日本語の文章です。

\[
x = a + b
\]

\end{document}
```

## ビルドとプレビュー

ctrl + alt + b (Mac では command(⌘) + option(⌥) + b) で TeX ソースのビルドができます。  
また、ctrl + alt + v (command(⌘) + option(⌥) + v) でプレビューが開きます。

# OverleafでLaTeXを書く方法



Product Solutions テンプレート Pricing Help & resources [Sign up](#) [ログイン](#)

## `\begin{reports}`

Write like a rocket scientist with Overleaf  
—the collaborative, online LaTeX editor that *anyone* can use.

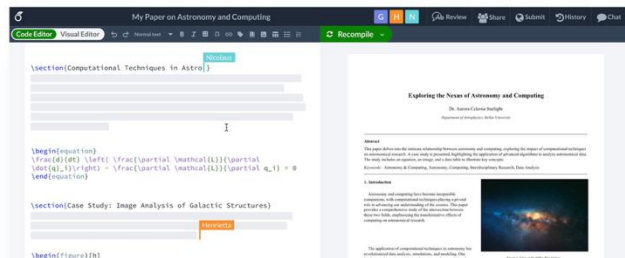
[Sign up with Google](#)

[Sign up with ORCID](#)

OR

[Sign up for free](#)

[I agree to our \[terms of service\]\(#\) and \[privacy notice\]\(#\).](#)



Product Solutions テンプレート Pricing Help & resources [プロジェクト](#) [アカウント](#)

## Overleaf によろこ

Create a new project

Learn LaTeX with a tutorial

Browse templates

~~空のプロジェクト~~

プロジェクト例

プロジェクトのアップロード

GitHubからインポート

The screenshot displays the Overleaf online LaTeX editor interface. On the left, the 'Code Editor' shows the source code for 'main.tex'. The code includes a document class, a package for graphics, title and author information, and a section titled 'Introduction'. The '\maketitle' command is highlighted in red, with a red arrow pointing to it. Below the code editor, the 'File outline' shows the 'Introduction' section. On the right, the rendered PDF output is shown, featuring the title '1 Introduction' and the date 'April 2026'.

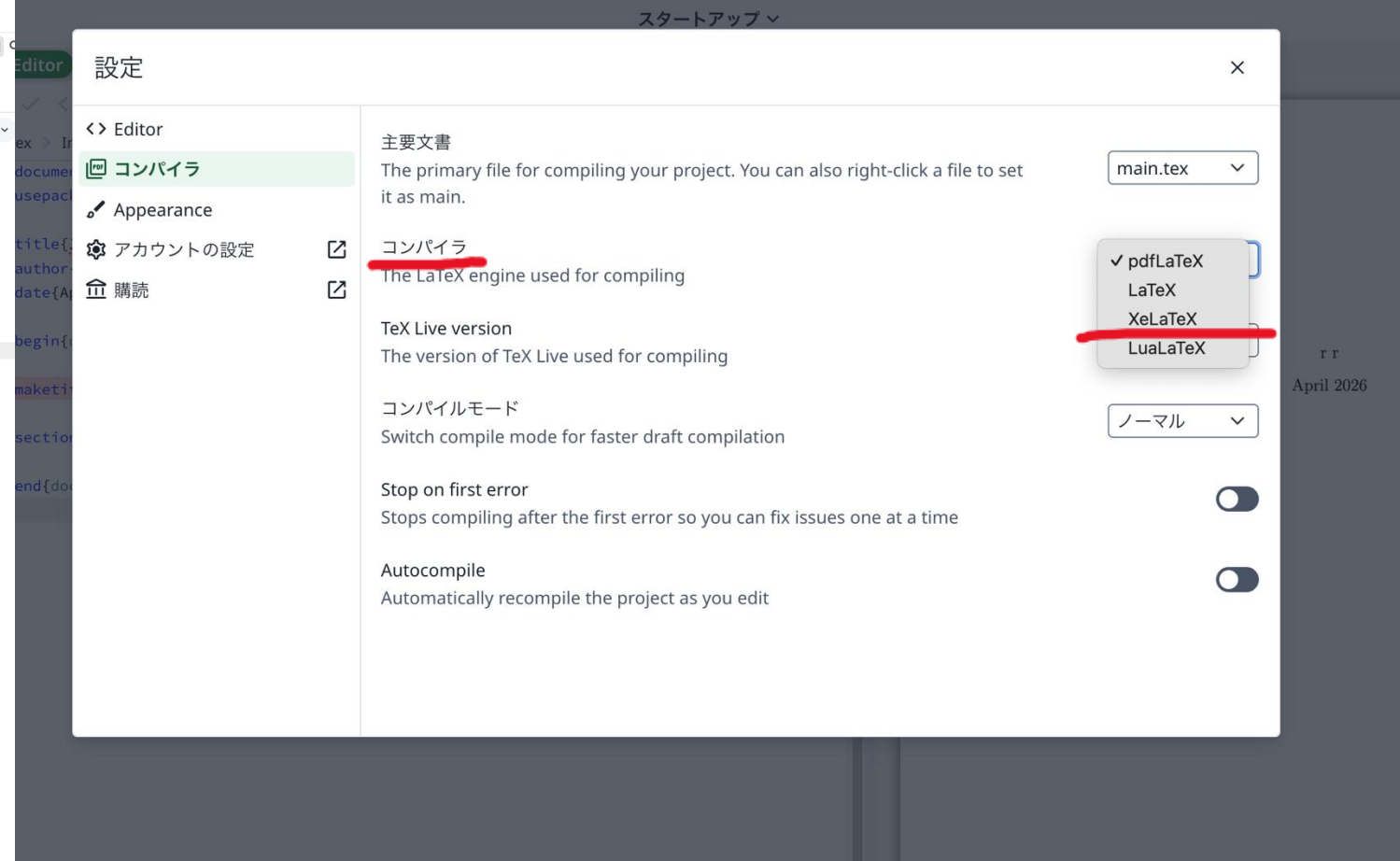
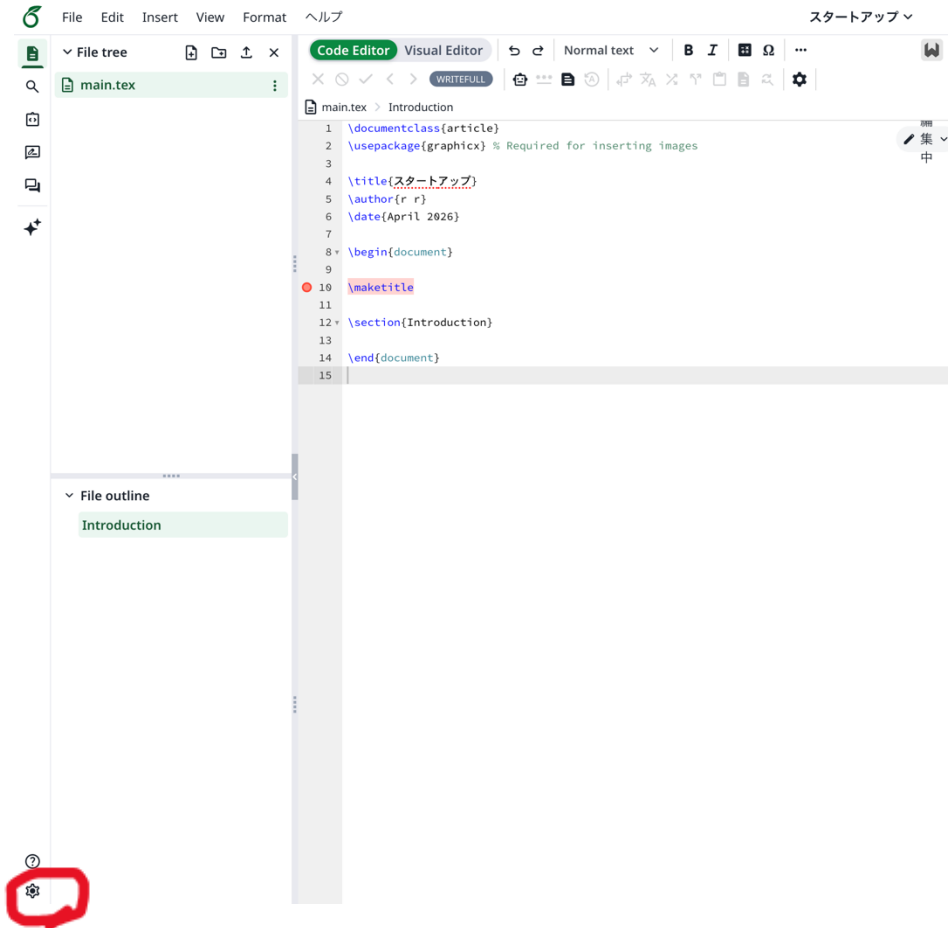
```
1 \documentclass{article}
2 \usepackage{graphicx} % Required for inserting images
3
4 \title{スタートアップ}
5 \author{r r}
6 \date{April 2026}
7
8 \begin{document}
9
10 \maketitle
11
12 \section{Introduction}
13
14 \end{document}
15
```

1 Introduction

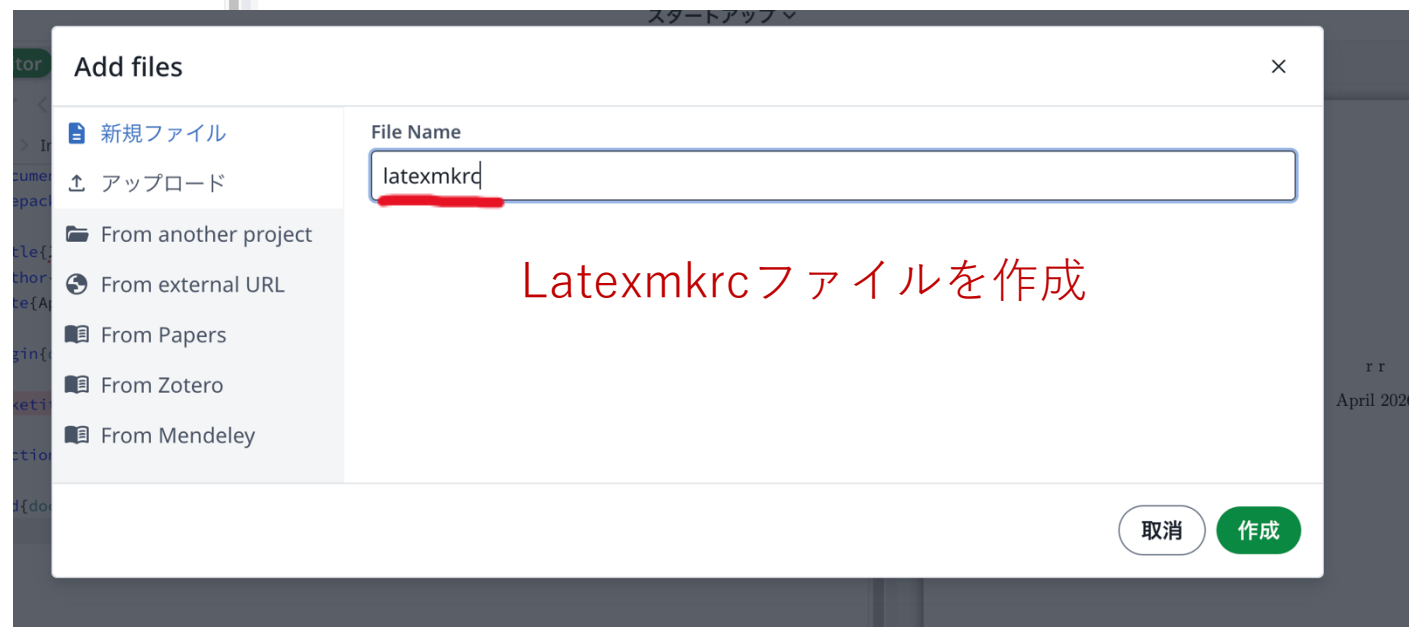
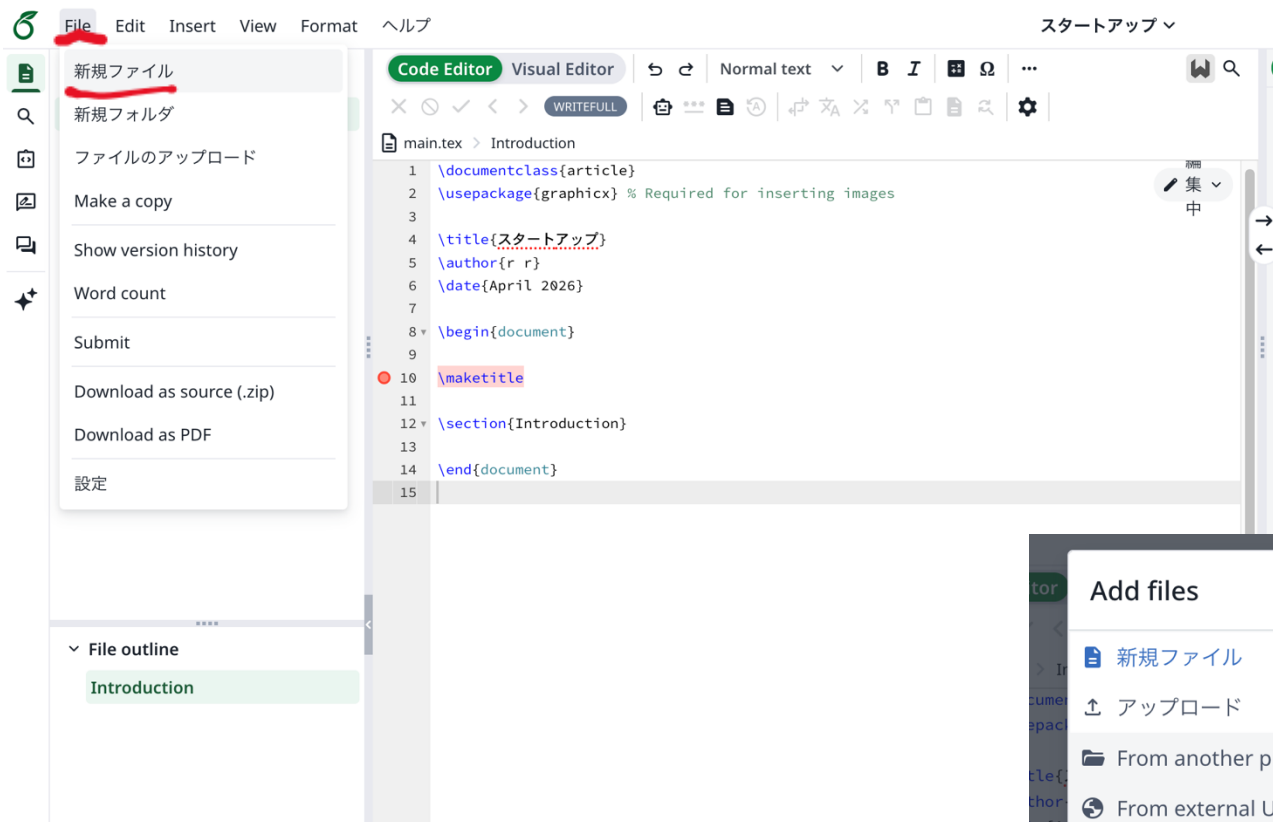
April 2026

Overleafでは、日本語を利用する際に事前の準備が必要

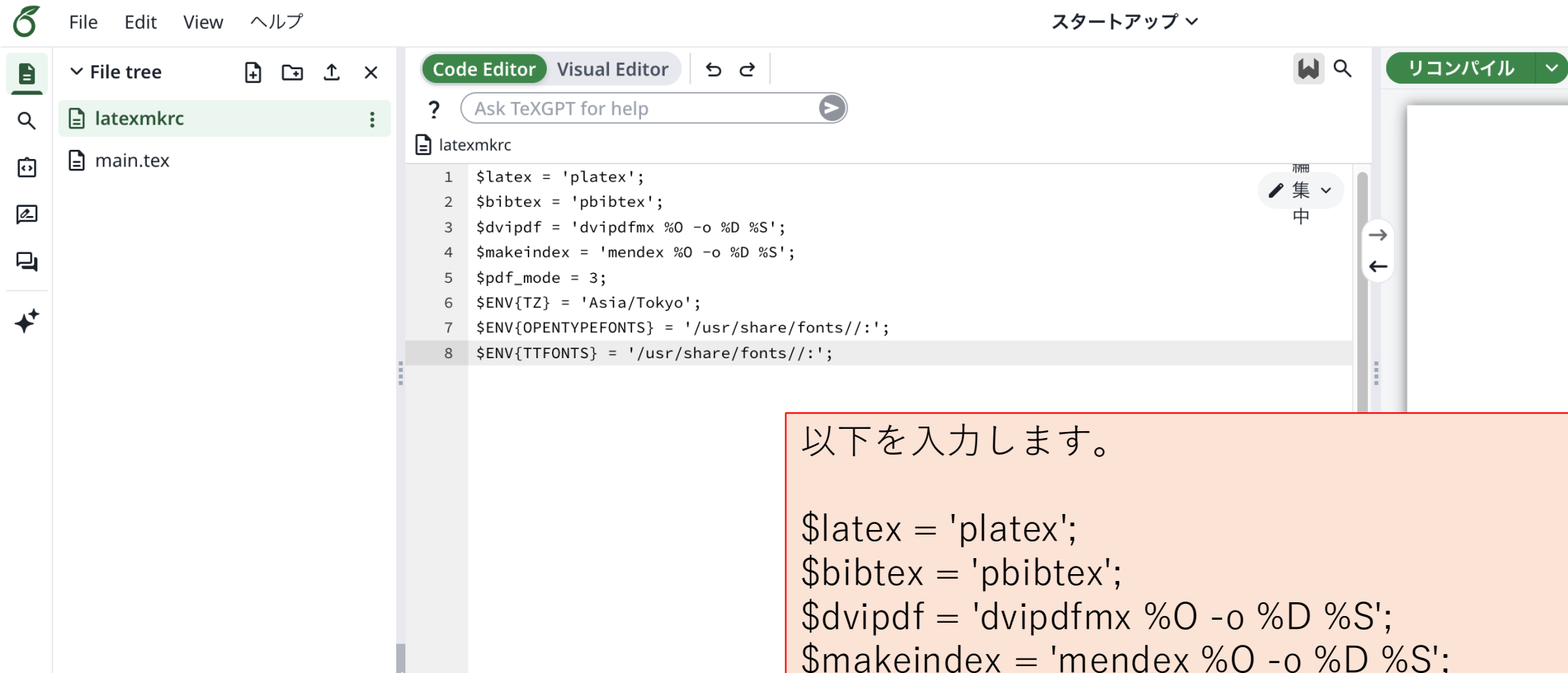
# Overleafの日本語設定



# Overleafの日本語設定



# Overleafの日本語設定



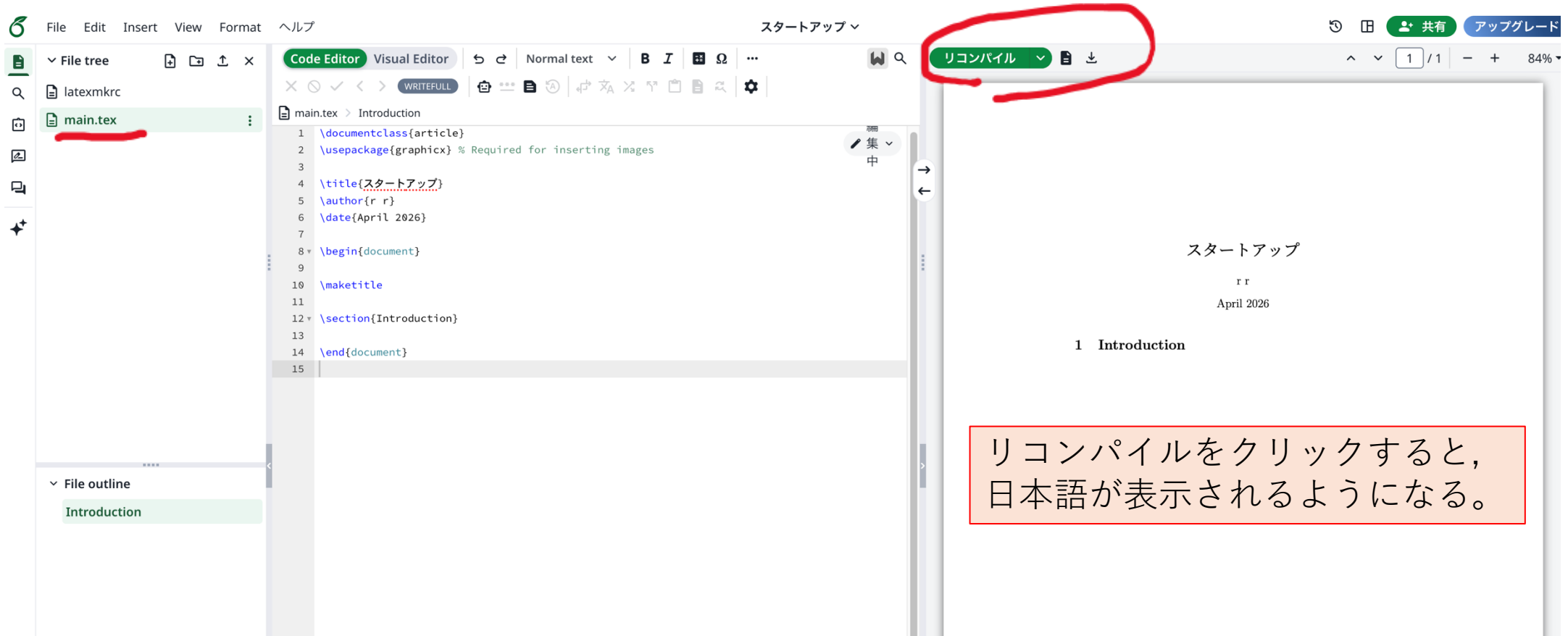
The screenshot shows the Overleaf web editor interface. The top menu bar includes 'File', 'Edit', 'View', and 'ヘルプ'. The main editor area is titled 'Code Editor' and shows the contents of the 'latexmkrc' file. The file content is as follows:

```
1 $latex = 'platex';  
2 $bibtex = 'pbibtex';  
3 $dviPDF = 'dviPDFmx %O -o %D %S';  
4 $makeindex = 'mendex %O -o %D %S';  
5 $pdf_mode = 3;  
6 $ENV{TZ} = 'Asia/Tokyo';  
7 $ENV{OPENTYPEFONTS} = '/usr/share/fonts//:~';  
8 $ENV{TTFFONTS} = '/usr/share/fonts//:~';
```

以下を入力します。

```
$latex = 'platex';  
$bibtex = 'pbibtex';  
$dviPDF = 'dviPDFmx %O -o %D %S';  
$makeindex = 'mendex %O -o %D %S';  
$pdf_mode = 3;  
$ENV{TZ} = 'Asia/Tokyo';  
$ENV{OPENTYPEFONTS} = '/usr/share/fonts//:~';  
$ENV{TTFFONTS} = '/usr/share/fonts//:~';
```

# Overleafの日本語設定



The screenshot shows the Overleaf editor interface. The top menu bar includes 'File', 'Edit', 'Insert', 'View', 'Format', and 'ヘルプ'. The 'Code Editor' tab is active, showing the LaTeX source code for 'main.tex'. The code includes commands for document class, packages, title, author, date, and document structure. The rendered output on the right shows the document's title 'スタートアップ', author 'r r', date 'April 2026', and a section '1 Introduction'. A red circle highlights the 'リコンパイル' (Recompile) button in the top right corner. A red box contains the text: 'リコンパイルをクリックすると、日本語が表示されるようになる。' (Clicking the 'Recompile' button will allow Japanese to be displayed.)

```
1 \documentclass{article}
2 \usepackage{graphicx} % Required for inserting images
3
4 \title{スタートアップ}
5 \author{r r}
6 \date{April 2026}
7
8 \begin{document}
9
10 \maketitle
11
12 \section{Introduction}
13
14 \end{document}
15
```

スタートアップ  
r r  
April 2026

1 Introduction





リコンパイルをクリックすると、  
日本語が表示されるようになる。

# ダウンロード方法

The screenshot displays the Overleaf web interface. At the top left, the Overleaf logo is circled in red. The interface is divided into several sections:

- File Tree:** Located on the left, it shows a project named 'main.tex' selected.
- Code Editor:** The central area shows the LaTeX source code for 'main.tex':

```
1 \documentclass{article}
2 \usepackage{graphicx} % Required for inserting images
3
4 \title{スタートアップ}
5 \author{r r}
6 \date{April 2026}
7
```
- Project List:** On the right, under 'あなたのプロジェクト' (Your Projects), there is a table with one project listed:

タイトル	管理者	最終変更 ↓	操作
スタートアップ	You	9 minutes ago by You	   

The download icon in the '操作' column of the project list is highlighted with a red underline.

## 1. 文章の基本

```
\documentclass[uplatex]{jsarticle}

\begin{document}

ここに文章を書く

\end{document}
```

## 2. 見出し

```
\section{大見出し}
\subsection{小見出し}
\subsubsection{さらに細かい見出し}
```

## 3. 数式

インライン(文章中)

```
 $x = a + b$ 
```

独立した数式

```
\[
x = a + b
\]
```

## 4. 箇条書き

### ・ 付きリスト

```
\begin{itemize}
\item りんご
\item みかん
\end{itemize}
```

### ・ 番号付き

```
\begin{enumerate}
\item 1つ目
\item 2つ目
\end{enumerate}
```

## 5. 図の挿入

```
\usepackage[dvipdfmx]{graphicx}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image.png}
\caption{サンプル図}
\end{figure}
```

## 6. 表の挿入

```
\begin{tabular}{c c}
A & B \\
1 & 2 \\
\end{tabular}
```

2020/4/15

## 7. 参照(図番号などを自動参照)

```
\label{fig:test}  
\ref{fig:test}
```

## 8. 強調

```
\textbf{太字}  
\textit{斜体}
```

## 9. コメント

```
% これはコメント
```

# 研究ノート

その日にした作業、調べた内容、解析したデータ、収集した論文、気がついたことなど記録に残します。

- ① 日付
- ② 研究のタイトル
- ③ 研究の目的：何を知りたいのか，得たいのかが明確になるように記載する。
- ④ 研究の手法や計画
- ⑤ 結果：どういう結果が得られたかを明確な文章で記載する。
- ⑥ 考察：実験結果の解釈，明らかにになったこと，わからなかったこと，実験の問題点，などを文章で書く。
- ⑦ 気づきや他者からのアドバイス
- ⑧ 計算過程など，ちょっとしたことも書き残しておく。
- ⑨ その他、次やることなど

# 文献管理

アクセス



J-STAGE

Google Scholar

CiNii Research

ScienceDirect®

論文管理ツール

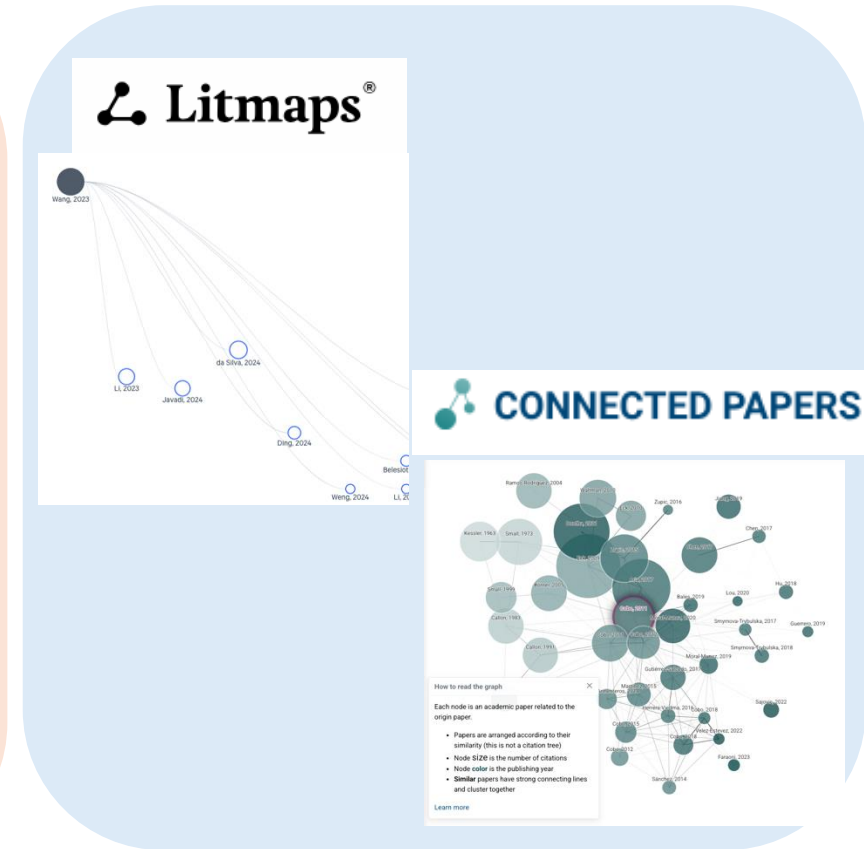


Zotero

EndNote

MENDELEY

論文mapping



Litmaps®

CONNECTED PAPERS

How to read the graph

Each node is an academic paper related to the origin paper.

- Papers are arranged according to their similarity (this is not a citation tree)
- Node SIZE is the number of citations
- Node color is the publishing year
- Similar papers have strong connecting lines and cluster together

Learn more