



**ROBOT SYSTEM DESIGN  
LABORATORY**

# 卒業論文執筆に向けた **LaTeX と GitHub**

発表者：小島 大輝

Jan 7, 2026

# 1. LaTeX とは

## TeX と LaTeX の歴史

- **TeX (ドナルド・クヌース):**  
組版品質に不満を持ち開発。数式の組版において事実上の標準。
- **LaTeX (レスリー・ランポート):**  
TeX を機能強化。文書の「論理構造」と「視覚的レイアウト」を分離。

## Word との違い（構造とデザインの分離）

「WYSIWYG<sup>a</sup>」ではなく、コマンドで構造を指定。

- `\section{はじめに}`と書くだけで、フォントサイズや配置が自動適用される。
- クラスファイルを変えるだけで、論文形式やプレビュー形式などレイアウトを一括変更可能。
- 煩わしい見た目の調整から解放される。

---

<sup>a</sup>what you see is what you get

## 2. LaTeX の環境構築

本輪講では **Visual Studio Code (VSCode)** を使用します。

### Windows における構築手順

- ① **TeX Live** のインストール: 公式サイトより実施.
- ② **VSCode** の導入: Japanese Language Pack も推奨.
- ③ 拡張機能「**LaTeX Workshop**」: 必須. コンパイルやプレビュー機能を提供.
- ④ **settings.json** の設定: pLaTeX + latexmk を使用するように記述.

### WSL/Ubuntu の場合 (補足)

`sudo apt install texlive-full` と VSCode の「Remote - WSL」拡張機能を使用することで、Linux コマンドを活用した執筆が可能。



### 3. LaTeX で執筆をする

- 章立て:  
`\chapter, \section, \subsection` 等で階層構造を記述.
- 数式:  
行内数式は  $y = ax + b$ , 別行立ては `equation` 環境など  
を使用.

#### BibTeX による参考文献管理

- 参考文献情報をネットから落とすだけで利用化.
- `.bib` ファイルに文献情報をリスト化.
- 本文中で `\cite{key}` とするだけで、自動的に番号振りとリス  
ト出力が行われる.

# 画像の並べ方

- 1枚だけの場合

```
\begin{figure}
  \centering
  \includegraphics[width=60mm]{picture/figure.png}
\end{figure}
```

- 2枚以上の場合

```
\begin{columns}
  \begin{column}{0.5\textwidth}
    \centering
    \includegraphics[width=\textwidth]{picture/figA.png}
  \end{column}
  \begin{column}{0.5\textwidth}
    \centering
    \includegraphics[width=\textwidth]{picture/figB.png}
  \end{column}
\end{columns}
```

# スニペットを使う

画像の設定など何度も書く煩わしい記述は vscode のスニペットを利用することで呼び出せるようになる。

## スニペットとは？

コーディング作業などでよく記述するテンプレ的な要素を保存しておき、いつでも呼びだせるようになる機能。

画像の挿入や部分的な2段組み、表の作成などはスニペットにしておくと良い。

# 4. 大規模な文書(卒業論文)を書く

## 分割コンパイルとディレクトリ構成

1つのファイルに全て書かず, \input 等で分割する.

- **main.tex**: 統合ファイル
- **sec/**: 本文 (intro.tex, method.tex...)
- **picture/**: 画像ファイル
- **include/**: 研究室のスタイルファイル

## 章立ての仕方

- 先輩の章立てを参考にすれば良い.
- IMRaD 方式

Introduction (背景), Methods (手法), Results (結果),  
Discussion (考察) の構成で論理的に記述する. 標準的な論文の構成.

# 5. GitHubで卒論を管理する

バージョン管理システムにより、卒論のセーブ・ロードが可能に！

## 基本サイクル

- ① git add . (変更対象を選択)
- ② git commit -m "セーブの名前" (変更を記録)
- ③ git push origin master (GitHubへ送信)

## 補足 : .gitignore の設定

コンパイルで生成される中間ファイル (.aux, .log, .fls 等) は管理対象外に設定すると便利。

データを戻したい時は github からコミット履歴を遡る。