

수학자를 위한 \LaTeX

Part 2: LyX for \LaTeX

권현우

서강대학교

Contents

1. $\text{T}_{\text{E}}\text{X}$ 을 쓸 때 무엇이 불편한가?
2. 한글 문서 작성을 위한 세팅
3. 주요 기능

T_EX을 쓸 때 무엇이 불편한가?

- 커맨드를 외우는게 힘들다 (이건 어쩔수 없고...)

T_EX을 쓸 때 무엇이 불편한가?

- 커맨드를 외우는게 힘들다 (이건 어쩔수 없고...)
- 프로그래밍 언어개념이 어렵다 (이것도 어쩔 수 없고...)

T_EX을 쓸 때 무엇이 불편한가?

- 커맨드를 외우는게 힘들다 (이건 어쩔수 없고...)
- 프로그래밍 언어개념이 어렵다 (이것도 어쩔 수 없고...)
- 글을 쓸 때 내가 무엇을 쓰고 있는지 바로 안 보인다!

T_EX을 쓸 때 무엇이 불편한가?

- 커맨드를 외우는게 힘들다 (이건 어쩔수 없고...)
- 프로그래밍 언어개념이 어렵다 (이것도 어쩔 수 없고...)
- 글을 쓸 때 내가 무엇을 쓰고 있는지 바로 안 보인다!

`$\int_a^b f(x) dx$`

$$\int_a^b f(x)dx$$

두 가지 방법

- Visual Studio Code
 - KaTeX (<https://katex.org/>)
 - LaTeX Workshop
 - Markdown+Math
- LyX

Visual Studio Code의 LaTeX Workshop

- Math Preview
- Cite Preview
- Picture Preview
- Snippet
 - BAL (Begin ALign)
 - BSAL (Begin Star ALign)
 - BFR (Begin FRame)
 - SSE (Start SEction)
 - ... (snippet)

Visual Studio Code의 LaTeX Workshop

적어도 Math Preview에 관한 한계

- 결과물이 바로 보이는 건 아니고 마우스로 가져다 대야한다.
- amsmath, amssymb을 넘어간 패키지를 사용할 경우 수식 출력에 제한적

```
\begin{align*}
\int_a^b f(x) dx &= \int_c^d f(x) dx \\
&\relphanto{=} \int_c^d f(x) dx \\
\mathbb{b}a
\end{align*}
 $\mathbb{b}a$ 
```

- 간단한 명령문으로 만들어진 코드를 잘 읽지만, 조건문을 이용하여 코딩한 경우는 작동하지 않음.

```
\@ifstar\@normb\@normi\bmug\apple\@ifstar\@n
\end{thm}
\end{thm}
```

Visual Studio Code의 Markdown+Math

- Markdown 문서를 작성할 때, 수식 파트는 KaTeX을 이용하여 출력
 - 마크업 언어: 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어의 한 가지 (html, \LaTeX , ...)
 - 마크다운: 2004년에 존 그루버와 아론 스위츠가 만든 마크업 언어. 다양한 포맷으로 변경하는 것을 지향
- Beamer 같은 것도 쉽게 만들 수 있긴 한데, 완벽히 호환된다고 하기에는 어렵기도...
- 커스텀 커맨드 제한적

LyX (1995–)



- WYSIWYM (What You See Is What You Mean), not WYSIWYG(What You See Is What You Get, 대부분의 워드프로세서)
- $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ 과 호환이 잘 되는 GUI 문서편집기
- 따라서 직관적인 $\text{T}_\text{E}\text{X}$ 문서작성이 가능하며, 각종 단축키를 사용함으로써 효율적인 문서작성이 가능

LyX, 언제 쓰면 좋을까?

1. 내용을 빠르게(!) 채워넣어야 할 때
2. Beamer와 같이 단순노동이 많이 필요한 경우
3. 레이텍 수식코드의 실현물을 빠르게 보고 싶을 때

한계

- 공동연구를 할 때는 비효율적
- 모든 패키지를 지원하는 것은 아니므로 사용할 수 있는 범위는 다소 제한적
- LyX 하나로 모든 걸 해결할 수 없다. 내용작성에만 신경쓸때 유용

한글 문서 작성을 위한 세팅

- Document-Settings-Language
 - Text Layout – Spacing Line spacing: custom (1.33)
 - Page Layout – Page Layout (Page style: default, Two-sided document)
 - Language – English
 - LaTeX Preamble – `\usepackage{kotex}`
- 컴파일 단축키 (ctrl+R 또는 command+R)

또는

- Document-Settings-Language
 - Fonts – Use non-TeX fonts (via XeTeX/LuaTeX) 클릭
 - Text Layout – Spacing Line spacing: custom (1.33)
 - Page Layout – Page Layout (Page style: default, Two-sided document)
 - Language – Korean
 - LaTeX Preamble – `\usepackage{kotex}`
- 컴파일 단축키 (ctrl+R 또는 command+R)

한글 문서 작성을 위한 세팅

```
\everymath\expandafter{\the\everymath\def\baselinestretch{1.0}\selectfont}  
\everydisplay\expandafter{\the\everydisplay\def\baselinestretch{1.0}\selectfont\abovedisplayskip  
  \belowdisplayskip 6.5pt plus 1pt minus 1pt}  
\allowdisplaybreaks
```

주요 단축키들

윈도우에 비해 맥은 단축키 작동이 잘 되지 않음. (맥의 경우에는 `command+shift-/`를 눌러서 방향키로 이동하기 또는 검색하기)

- 행중수식 작성 (`ctrl+M` 또는 `command+R`)
- 별행수식 작성 (`ctrl+shift+M` 또는 `command+shift+M`)
- 수식에서 여러 행 만들기 (`ctrl+Enter` 또는 `command+Enter`)
- LaTeX 코드 입력 (`ctrl+L` 또는 `command+L`)
- 라벨 입력 (`alt+l+l`)
- 상호참조 입력 (`alt+l+r`)
- 그 외는 클래스에 따라 다름

세부 사용방법: 모듈

\LaTeX 의 package를 불러오거나 패키지의 옵션을 불러오는 기능

- Theorems
 - Theorems (AMS) / Theorems (Numbered by Section)
- Subequations
- Number Equations by Section (Figures, Tables)
- Customisable Lists (enumitem)

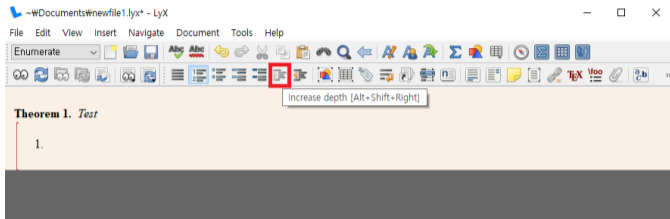
Theorem

Theorem 1. *Test*

Theorem 2. Additional Theorem Text aa *Test2*

- Alt+P+Enter (맥: Control+P+Enter) : 새로운 환경으로 시작
- Alt+A+1 (맥: Control+A+1) : 환경의 추가옵션 지정

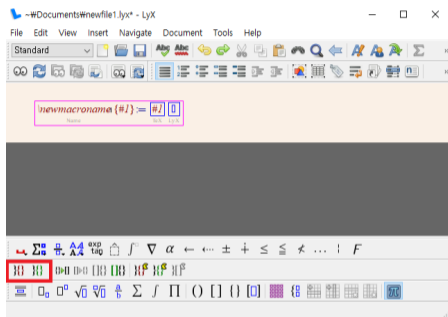
Theorem




```
\begin{theorem}  
\begin{enumerate}  
\item  
\item  
\end{enumerate}  
\end{theorem}
```

과 같이 작성하기 위해서 사용하는 옵션

Custom command



-  버튼으로 명령어 제작 가능
- 간단한 명령어 수준을 만들 수 있음
- LyX코드에는 LyX에서 출력할 모양을 보여줌.

공통: 잠시 \LaTeX 프로그래밍을 봅시다

```
\newcommand{\mycommand}[2]{ #1 #2}
```

```
\renewcommand{\mycommand}[2]{#1 #2}
```

```
\newcommand{\bZ}{\mathbb{Z}}
```

```
\newcommand{\norm}[1]{\left\| #1 \right\|}
```

커맨드를 정의하기 위한 방법: 파라미터 이해

파라미터는 매크로에 개수를 넣는 것이다.

```
\newcommand{이름}[파라미터개수]{내용}
```

예시:

```
\newcommand{\norm}[2]{\Vert #1 \Vert_{#2}}
```

LyX의 주요 기능

- Insert - File - LyX Document
- Import 기능이 있긴 하지만, 호환성이 어쩔수 없이 부족함
- Export는 xetex 또는 luatex으로 출력하는 것이 좋음.

Thank you for your attentions!