

L^AT_EX 설치하기

- ★ T_EX은 여러 가지 배포판이 있습니다.
- ★ T_EX Live 배포판 설치하기 (가장 쉬운 방법)
 - ★ <https://www.tug.org/texlive/>에 방문하여 설치
 - ★ `texlive-full`을 받으면 정신건강에 매우 편하지만, 약 **4.9 GB**를 차지합니다.
 - ★ `texlive-basic`을 받으면 초기 용량을 **64 MB**까지 줄일 수 있지만, 필요한 패키지를 일일이 설치해야 합니다.
- ★ MikTeX (Windows) (<https://miktex.org/>)
- ★ TnXTeX (<http://wiki.ktug.org/wiki/wiki.php/TnXTeX>)
- ★ TinyTeX (<https://yihui.name/tinytex/>)

MikTeX은 T_EX Live 기반 배포판과 설정 방법이 달라 추천드리지 않습니다.

T_EX Live 패키지 설치하기 (스킵 가능)

- ★ `texlive-full`이 아닌 더 적은 패키지를 포함하고 있는 버전을 받으셨다면, 추가로 설치해야 하는 패키지들이 있을 수 있습니다.
- ★ 문서를 작성하다가 File ``~~~.sty' not found`.라는 에러 메시지가 발생하면, 다음 명령어를 터미널 (혹은 cmd)에 입력해 주세요.
 - ★ `tlmgr search --global --file "/패키지_이름.sty"`를 입력하면
 - ★ ‘패키지_묶음_이름: `texmf-dist/.../패키지_이름.sty`’라는 문구를 볼 수 있고,
 - ★ `tlmgr install 패키지_묶음_이름`을 치면 설치됩니다.
- ★ 다운로드 받은 `sty` 파일을 설치하고 싶다면,
<https://github.com/msquare-kaist/mathletter-package/blob/master/documents/manual.pdf>를 참조해 주세요.

T_EX이란?

- ★ T_EX: Donald Knuth가 개발한 문서 조판 도구
- ★ L_AT_EX: Leslie B. Lamport가 개발한 T_EX의 확장 (매크로 모음)
 - ★ L_AT_EX 2_ε: 가장 많이 쓰이는 L_AT_EX 버전
 - ★ L_AT_EX 3: 현재 개발 중인 L_AT_EX 버전
- ★ 이외에도 ConT_EXt이라는 확장이 있습니다.
- ★ Overleaf: 가장 유명한 온라인 L_AT_EX 동시 편집 서비스
(<https://overleaf.com>)

Token

- ★ $\text{T}_\text{E}\text{X}$ 문서는 토큰들로 이루어져 있으며, $\text{T}_\text{E}\text{X}$ 엔진이 각 토큰을 **전개** (expand) 하면서 실행됩니다.
- ★ **매크로** (macro) 또는 **정의** (definition) 란 기본적으로 정의되어 있는 토큰들 (primitives) 을 이용하여 정의된 토큰들을 말합니다.
- ★ 매크로는 보통 `\macro` 와 같이 백슬래시로 시작하고 알파벳이 뒤에 따라옵니다.
- ★ 한 글자짜리 매크로는 알파벳이 아니어도 올 수 있습니다. `\` , `\#` , ...

문서 구조

- ★ `\documentclass[a4paper,11pt]{article}` 문서의 논리적 구조를 지정합니다.
 - ★ book, report, memoir 등의 class가 있습니다.
- ★ `\usepackage{amsmath}` 미리 정의된 매크로를 **패키지**에서 가져옵니다.
 - ★ 사실은 `\usepackage` 는 `\RequirePackage` 로 정의됩니다.
- ★ `\title{...}` , `\author{...}` , `\date{...}`
- ★ `\begin{document}` 문서를 시작합니다.
- ★ `\maketitle` 제목, 저자, 날짜를 출력합니다.
- ★ `% 주석입니다` 퍼센트 기호 뒤는 무시됩니다.
- ★ `\end{document}` 문서를 끝냅니다.

Sample Document

따라만 하세요 따라만 하세요

```
\documentclass[a4paper,10pt]{article}
\usepackage{amsmath}
\usepackage{kotex}
\title{제목}
```

```
\begin{document}
\maketitle
```

안녕하세요? \LaTeX 을 배우면 좋은 일이 있을 거예요. % 아닐 수도..

```
\makeatletter
당신이 입력했던 제목입니다: \@title.
\makeatother
\end{document}
```

글씨체

일반 텍스트의 글씨체

- ★ `{\normalfont}` 기본 폰트} 기본 폰트
- ★ `\textbf{굵게}` , `{\bfseries}` 굵게} **굵게**
- ★ `\textit{이탤릭}` , `{\itshape}` 이탤릭} *이탤릭*
 - ★ `\textit` 는 그 다음에 똑바로 쓴 글자가 왔을 때 간격 보정을 해줍니다.
 - ★ `\textit{(이탤)}릭` , `{\itshape (이탤)}릭`
italic
★ *italic*
- ★ `\textrm{명조}` , `{\rmfamily}` 명조} 명조
- ★ `\textsf{돋움}` , `{\sffamily}` 돋움} 돋움
- ★ `\textsl{기울임}` , `{\slshape}` 기울임} 기울임
- ★ `\textsc{SmallCaps}` , `{\scshape}` SmallCaps} SmallCaps
- ★ `\texttt{typewriter}` , `{\ttfamily}` typewriter} typewriter

글씨체

수식 모드의 글씨체

- ★ `\mathbf{Bold}` **Bold**
- ★ `\boldsymbol{\alpha}` α (알파벳이나 숫자가 아닌 기호를 굵게)
- ★ `\mathit{italic}` *italic*
- ★ `\mathrm{Roman}` Roman
- ★ `\mathsf{Sans-serif}` Sans – serif
- ★ `\mathbb{BLACK}` BLACK
- ★ `\mathcal{CALI}` *CALI*
- ★ `\mathscr{SCRIPT}` *SCRIPT*
- ★ `\mathfrak{Fraktur}` Fraktur

글씨 크기

★ `\tiny \scriptsize \footnotesize \small \normalsize`
`\large \Large \LARGE \huge \Huge`

★ AAAAAAAAAA

★ `\fontsize{글씨 크기}{line height}\selectfont`

★ 보통 line height는 글씨 크기의 1.2 배

수식 입력하기

- ★ `$... $` 다른 글씨들과 같이 출력되는 수식입니다. 예) $\alpha^2\beta$
 - ★ `\(... \)` 을 쓸 수도 있지만, 이 매크로는 **fragile**한 매크로라 다른 매크로의 인자로 들어가면 오류를 내므로 사용하지 않는 편이 낫습니다.
- ★ `\[... \]` 한 줄 전체를 차지하는 수식입니다.
 - ★ `$$... $$` 을 쓸 수도 있지만, 이 매크로는 여백 조절이나 수식 넘버링, 또 에러 메시지 출력 등에서 위의 것보다 안 좋습니다.
- ★ 기본적인 수식 매크로:
 - `$ \alpha, \beta, \frac{\sqrt{2}}{3}, \dots $`
 - ★ $\alpha, \beta, \frac{\sqrt{2}}{3}, \dots$
- ★ 더 많은 매크로와 기호는
 - ★ <https://www.codecogs.com/latex/eqneditor.php>나
 - ★ <http://detexify.kirelabs.org/classify.html>에서!

수식 매크로는 많이 외우셨으면 좋겠습니다

```
\documentclass[a4paper,10pt]{article}
\usepackage{amsmath,amsfonts,amssymb,mathrsfs,mathtools,kotex}
\title{\scshape Practice}
\begin{document}
\maketitle
\[ \text{근의 공식}: \quad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad ]
 $\alpha$ ,  $\beta$ 는 있지만  $\texttt{\char`\\Alpha}$ ,  $\texttt{\char`\\Beta}$ 는 없다!

Radical of an ideal  $\sqrt{\mathfrak{a}}$ 

크기가 조절되는 괄호를 열 때에는  $\left( \frac{a}{b} \right)$ 처럼 씁니다.
강의할 때 어느 정도 설명해주세요  $\verb!^^!$ 
\end{document}
```

^^

공백, 행간 간격

- ★ 문단을 띄울 때에는 엔터를 두 번 치거나 `\par` 를 입력합니다.
- ★ 강제로 한 줄을 띄울 때에는 `\\` 를 입력합니다.

공백, 행간 간격

- ★ 문단을 띄울 때에는 엔터를 두 번 치거나 `\par` 를 입력합니다.
- ★ 강제로 한 줄을 띄울 때에는 `\\` 를 입력합니다.
- ★ `\! \, \quad \qquad ~` 는 텍스트 모드에서의 공백을 나타냅니다.
크기는 다음과 같습니다.
 - ★ `a a\!a\,a\quad a\qquad a~a` → a aa a a a a
 - ★ `\!` 는 간격을 줄입니다.
 - ★ `~` 는 끊어지지 않는 공백을 나타내는 ‘active character’입니다.
- ★ `\! \, \> \: \; \quad \qquad ~` 는 수식 모드에서의 공백을 나타냅니다. (직접 테스트 해보세요)

공백, 행간 간격

- ★ 문단을 띄울 때에는 엔터를 두 번 치거나 `\par` 를 입력합니다.
- ★ 강제로 한 줄을 띄울 때에는 `\\` 를 입력합니다.
- ★ `\! \, \quad \qquad ~` 는 텍스트 모드에서의 공백을 나타냅니다.
크기는 다음과 같습니다.
 - ★ `a a\!a\,a\quad a\qquad a~a` → a aa a a a a
 - ★ `\!` 는 간격을 줄입니다.
 - ★ `~` 는 끊어지지 않는 공백을 나타내는 ‘active character’입니다.
- ★ `\! \, \> \: \; \quad \qquad ~` 는 수식 모드에서의 공백을 나타냅니다. (직접 테스트 해보세요)
- ★ `\hspace{길이}` , `\hspace*{길이}` 는 가로로 <길이>만큼의 간격을 만듭니다. 원래는 한 줄을 시작할 때엔 무시되는데, *이 붙으면 살아납니다.
- ★ `\vspace{길이}` , `\vspace*{길이}` 는 세로 버전

공백, 행간 간격

- ★ 문단을 띄울 때에는 엔터를 두 번 치거나 `\par` 를 입력합니다.
- ★ 강제로 한 줄을 띄울 때에는 `\\` 를 입력합니다.
- ★ `\! \, \quad \qquad \sim` 는 텍스트 모드에서의 공백을 나타냅니다.
크기는 다음과 같습니다.
 - ★ `a a\!a\,a\quad a\qquad a\sim a` → a aa a a a a
 - ★ `\!` 는 간격을 줄입니다.
 - ★ `\sim` 는 끊어지지 않는 공백을 나타내는 ‘active character’입니다.
- ★ `\! \, \> \: \; \quad \qquad \sim` 는 수식 모드에서의 공백을 나타냅니다. (직접 테스트 해보세요)
- ★ `\hspace{길이}` , `\hspace*{길이}` 는 가로로 <길이>만큼의 간격을 만듭니다. 원래는 한 줄을 시작할 때엔 무시되는데, *이 붙으면 살아납니다.
- ★ `\vspace{길이}` , `\vspace*{길이}` 는 세로 버전
- ★ `\noindent` 는 문단 시작 시 들여쓰기를 없앱니다.

달러나 백슬래시는 어떻게 입력할까요

미국인이 만들었는데 달러를 입력 못할 리가 없겠죠!!

★ `\& \% \$ \# _ \{ \} \char`~_\char`~_\char`~\`

★ `& % $ # _ { } ~ ^ \`

달러나 백슬래시는 어떻게 입력할까요

미국인이 만들었는데 달러를 입력 못할 리가 없겠죠!!

★ `\& \% \$ \# _ \{ \} \char`~_\char`~_\char`\\`

★ `& % $ # _ { } ~ ^ \`

★ 살짝씩 다르지만 다른 입력 방법도 많습니다:

★ `\textasciitilde_\textasciicircum_\textbackslash`

★ `~ ^ \`

달러나 백슬래시는 어떻게 입력할까요

미국인이 만들었는데 달러를 입력 못할 리가 없겠죠!!

★ `\& \% \$ \# _ \{ \} \char`~_\char`~_\char`~\`

★ `& \% $ \# _ \{ \} ~ ^ \`

★ 살짝씩 다르지만 다른 입력 방법도 많습니다:

★ `\textasciitilde_\textasciicircum_\textbackslash`

★ `~ ^ \`

★ `\^{} \~{} \string^ \string~ \string\`

★ `^ ~ ^ ~ \`

환경

Environments

★ `\begin{ENVNAME} ... \end{ENVNAME}` 꼴의 매크로를 모두 환경이라고 부릅니다.

★ 사실 이것은 `\ENVNAME ... \endENVNAME` 로 정의됩니다.

★ 여러 중요한 환경들:

★ `itemize`와 `enumerate` (with `enumitem`)

★ `matrix`

★ `amsthm`

★ `align`

★ `figure`와 `center`

★ `tabular`

itemize와 enumerate

예시만 넣어 놓으면 닥님께서 설명해 주실 거예요

enumitem 패키지를 불러온 뒤,

```
\begin{itemize}
  \item 첫 번째
  \item[2] 두 번째
  \item[ $\gamma$ ] wow
\end{itemize}

\begin{enumerate}[label={\roman*\arabic*}]
  \item 자동으로
  \item 올라가는
  \item 번호
\end{enumerate}
```

matrix

예시만 넣어 놓으면 쌤님께서 설명해 주실 거예요

수식 모드 안에서,

```
\begin{itemize}
\[
\begin{pmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{pmatrix} \cdot
\begin{bmatrix}
b_{11} & b_{12} \\
b_{21} & b_{22}
\end{bmatrix} =
\begin{vmatrix}
c_{11} & c_{12} \\
c_{21} & c_{22}
\end{vmatrix} \cdot \mathbf{I}
\]
```

amsthm

예시만 넣어 놓으면 닥님께서 설명해 주실 거예요

amsthm 패키지를 불러온 뒤,

```
\newtheorem{thm}{Theorem} % basic usagae
\newtheorem{lem}[thm]{Lemma} % shares the numbering w/ thm

\theoremstyle{definition}
\newtheorem{defn}[section]{Definition} % Def section.def_number

\theoremstyle{plain} % recover the theoremstyle
\newtheorem*{cor}{Corollary} % no numbering (*)

\begin{lem}[Lemma Name] This lemma is very awesome. \end{lem}
\begin{proof} Leave as an exercise. \end{proof}
```

figure와 center

예시만 넣어 놓으면 닥님께서 설명해 주실 거예요

float, graphicx, (adjustbox) 패키지를 불러온 뒤,

```
% H means that it tries to put on the exact position
\begin{figure}[H]
  \begin{center}
    \includegraphics[width=0.5\textwidth]{msquare}
    \caption{Gorgeous}
  \end{center}
\end{figure}
```


tabular

예시만 넣어 놓으면 닥님께서 설명해 주실 거예요

<https://www.tablesgenerator.com/>

매크로

귀찮은 일은 싫어하는 사람들의 모임

★ 편집부장님: “레이텍으로 구구단을 만들어 주세요!”

★ 안타까운 사람들: $2 \times 2 = 4$, $2 \times 3 = 6$, ...

매크로

귀찮은 일은 싫어하는 사람들의 모임

★ 편집부장님: “레이텍으로 구구단을 만들어 주세요!”

★ 안타까운 사람들: $\$2 \times 2 = 4\$$, $\$2 \times 3 = 6\$$, ...

★ 매크로를 배운 후:

```
\newcounter{left} \newcounter{right}
\forloop{left}{2}{\value{left} < 10}{%
  \noindent\forloop{right}{2}{\value{right} < 10}{%
    $\arabic{left} \times \arabic{right} =
      \the\numexpr\arabic{left} * \arabic{right}\relax$,
  }
\par
}
```

기본적인 매크로

plain T_EX, L^AT_EX 2_ε, L^AT_EX 3 버전

★ `\def\mymacro#1#2#3{\textit{#1}#2\textrm{#3}}`

★ 안타까운 사람들: `$2 \times 2 = 4$`, `$2 \times 3 = 6$`, ...

기본적인 매크로

plain T_EX, L^AT_EX 2_ε, L^AT_EX 3 버전

★ `\def\mymacro#1#2#3{\textit{#1}#2\textrm{#3}}`

★ 안타까운 사람들: `$2 \times 2 = 4$`, `$2 \times 3 = 6$`, ...

★ 매크로를 배운 후:

```
\newcounter{left} \newcounter{right}
\forloop{left}{2}{\value{left} < 10}{%
  \noindent\forloop{right}{2}{\value{right} < 10}{%
    $\arabic{left} \times \arabic{right} =
      \the\numexpr\arabic{left} * \arabic{right}\relax$,
  }
\par
}
```