

# LaTeX 활용하기

김 강 수\*

2008년 11월 28일

## 요약

2008년 11월 28일, 숭실대학교 교수학습센터 워크숍 학습 자료. LaTeX을 어느 정도 알고 계신 분을 대상으로, TeX과 LaTeX의 기본 사항을 상기시키고, 중급 활용자가 되기 위해 필요한 사항들을 실습과 함께 소개한다.

## 차 례

차 례	1
1 TeX: What and Why	3
1.1 TeX의 세 가지 측면	3
1.2 TeX이라는 말의 용법	3
1.3 글쓰기 도구로서의 LaTeX	3
1.4 LaTeX 문서 작성에 입문	4
2 첫 번째 난관, 설치	5
2.1 KC2008, 소개	5
2.2 설치에 관한 지침	5
2.3 참언	5
3 첫 문서를 작성해보자	6
3.1 미리 알아야 할 것	6
3.2 라텍 문서의 구성 요소	7
3.3 좀 긴 문서	7
4 중급자료의 첫걸음	9
4.1 라텍 사용하기의 두 모드	9
4.2 패키지의 활용	9
4.3 자신만의 레이아웃을 만들어보자	9
4.4 LaTeX 프로그래밍을 해보자	10

---

\*KTUG.collection@ktug.or.kr

4.5	폰트 문제 . . . . .	11
4.6	T <sub>E</sub> X 언어? . . . . .	12
4.7	또하나의 L <sup>A</sup> T <sub>E</sub> X을 쓸 이유 — Beamer . . . . .	13
5	Short Cuts . . . . .	14
5.1	매뉴얼과 샘플 . . . . .	14
5.2	수식 입력 익히기 . . . . .	14
6	L <sup>A</sup> T <sub>E</sub> X 문서 작성의 즐거움 . . . . .	15



## 1 T<sub>E</sub>X: What and Why

### 1.1 T<sub>E</sub>X의 세 가지 측면

**프로그래밍 언어** T<sub>E</sub>X은 D. Knuth가 만든 조판 프로그래밍 언어이다.

**컴파일러** T<sub>E</sub>X은 T<sub>E</sub>X 언어로 입력된 원본(source) 파일을 처리하여 오브젝트 파일을 얻어내는 컴파일러 프로그램이다.

**매크로 언어** T<sub>E</sub>X은 사용자가 다양한 수준에서 정의하고 활용할 수 있도록 하는 매크로 집합이다. T<sub>E</sub>X 매크로의 예로는 plainT<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt 등이 있다.

### 1.2 T<sub>E</sub>X이라는 말의 용법

**엔진, engine** T<sub>E</sub>X 매크로 언어의 가장 기본이 되는 primitive와 그 구현을 함께 엔진이라고 부른다. 현재 가장 많이 사용되는 엔진은 pdfT<sub>E</sub>X, 또는 pdf- $\epsilon$ -T<sub>E</sub>X.

**매크로, macro** 미리 정의된 매크로 세트로서 사용자가 T<sub>E</sub>X 언어에 익숙하지 않더라도 약속된 매크로 명령을 이용해서 원하는 결과를 더 쉽게 얻게 만드는 일종의 라이브러리이다. L<sup>A</sup>T<sub>E</sub>X.

**구현, implementation** T<sub>E</sub>X을 실제 활용하려면 방대한 폰트와 유틸리티가 필요하다. 이것을 하나의 시스템으로 묶어 해당 플랫폼에서 쉽게 운영할 수 있게 도와주는 T<sub>E</sub>X 관련 프로그램 모음으로 이루어진 시스템. KC2008, T<sub>E</sub>X Live, MiK<sub>T</sub><sub>E</sub>X. 배포판, 또는 설치 프로그램.

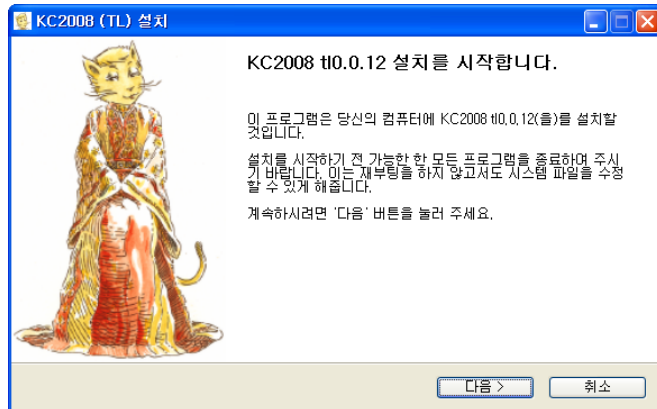
### 1.3 글쓰기 도구로서의 L<sup>A</sup>T<sub>E</sub>X

- L<sup>A</sup>T<sub>E</sub>X은 문서작성시스템(Document Preparation System), 즉, 글쓰기 도구이다. 그런 면에서 흔히 Word Processor와 비교된다.
- 주로 과학-기술 문헌의 글쓰기에 주로 활용된다. 논문, 학회지 기고, 단행본 출판 등.

- $\LaTeX$ 의 글쓰기는 <Mark-up>이다. 사용자는 조판 표지와 기타 문서의 구조(structure)와 외양(layout)을 결정하는 지시사항을 control sequence로 텍스트 사이에 써넣는다. 이런 방식의 글쓰기는 장점과 단점을 동시에 가지고 있다.
- $\LaTeX$ 의 글쓰기는 ‘programmable’하다. 즉 일정한 규칙에 따라서 문서 내의 각 요소에 대하여 프로그래밍을 적용할 수 있다.
- $\LaTeX$ 의 글쓰기는 ‘구조화 글쓰기’이다. 저자의 사유의 흐름을 구조화하고 내용에 집중할 수 있도록 해준다.
- $\LaTeX$ 이 채택하고 있는  $\TeX$ 의 수식 표현은, 적어도 수식 표현에 관한 한 가장 폭넓게 받아들여지고 있으며 익숙해진 사람에게 매우 효율적인 표현이 가능하다.

#### 1.4 $\LaTeX$ 문서 작성에 입문

- $\LaTeX$ 은 (좋은) 도구일 뿐, 저자의 사유를 대체하지 못한다.
- 그러나, 좋은 사유를 가장 잘 표현해주는 데 탁월하다.
- 마크-업 글쓰기의 장점을 최대한 활용하기로 결심해야 한다.
- $\LaTeX$ 이 “가장 좋은” 도구는 아닐지 모르나, 아주 좋은 도구임에는 틀림없다. 다만, 익숙해졌을 때, 익숙해질 때까지 약간의 학습은 필수적이다.



## 2 첫 번째 난관, 설치

### 2.1 KC2008, 소개

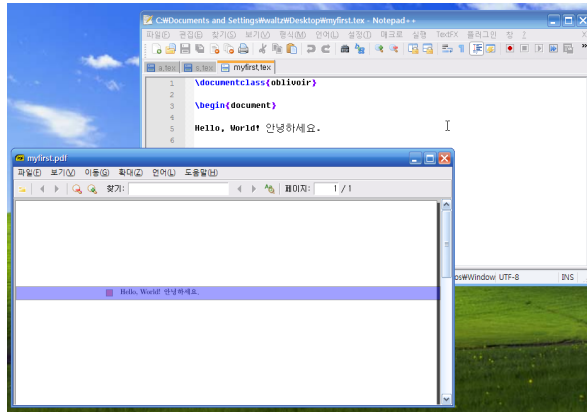
- KC2008은 KTUG과 KTS에서 공식적으로 배포하는 TeX 운영 시스템.
- KTUG이 설립된 2002년경부터 사용자 친화적인 편리한 (한글)TeX 시스템을 제공하려는 노력이 시작되었음.
- 2006년까지, MiKTeX-KTUG, HPack 등 다양한 시도들이 있었음.
- 2006년부터, KTUG Collection이라는 이름으로 단 한 번의 설치로 설정이 완료되고 즉시 문서 작성이 가능한 시스템을 개발해왔음.
- KC2008은 TeX Live를 기반 시스템으로 하여 한글환경과 다양한 유틸리티를 추가하고 전용 편집기를 제공함.
- 안정적이고 대단히 효율적인 실행 파일을 갖춘 국제 표준에 부합하는 TeX 운영 시스템.
- 문제가 생겼을 때 즉시 대응할 수 있고 사용자의 요구에 민감하게 반응하는 것이 최대 장점.

### 2.2 설치에 관한 지침

<http://www.ktug.or.kr/>의 "DOWNLOAD/Install" 메뉴. 또는 이기황 선생이 금년 공주대학교 워크숍에서 발표한 자료, <http://faq.ktug.or.kr/faq/LaTeXWorkshop/2008Fall?action=download&value=kc08inst.pdf> 참조.

### 2.3 참언

2000년대 중반까지만 해도, TeX 사용의 첫 난관이 설치였다. 그러나 KTUG Collection이 배포되고 있는 현재, 설치가 문제가 되는 경우는 극히 드물어졌고 복잡한 설정 과정도 거의 해결되었다. 예전에 TeX을 사용하시던 분들께 오늘날 설치 문제에 관한 한 어느 정도 극적인 변화가 생겼는지를 직접 물어보시면 재미있을 것임.



### 3 첫 문서를 작성해보자

#### 3.1 미리 알아야 할 것

```

\documentclass{oblivoir}%article, book, report, class,...
% 주석문. 이 부분이 preamble이다.

\begin{document}

안녕하세요. Hello, World!

\end{document}
    
```

- (1)  $\LaTeX$  문서는 `\documentclass` 선언으로 시작한다.
- (2)  $\TeX$ 의 제어명령(control sequences)은 역슬래시(`\`)로 시작하여 공백으로 끝난다.
- (3) `documentclass`에는 적절한 문서 클래스를 선언한다.
- (4) 전문(preamble)에서 `\usepackage` 문을 통하여 스타일을 로드하고 사용자 정의 설정을 행한다.
- (5) 본문은 `\begin{document}`와 `\end{document}` 사이에 온다.
- (6) 다음 문자들은 제어명령으로 쓰이므로 직접 입력하지 못한다. `& # ~ _ ^ \`이 문자들을 입력하려면 특별한 방법을 써야 한다.
- (7) 제어명령은 ‘command’라고 부르는 것과 ‘environment’라고 부르는 것이 있다.
- (8)  $\LaTeX$  원본 파일은 plain text (UTF-8) 파일이다. 특히 한글은 유니코드 UTF-8으로 저장되도록 주의를 기울이자. 좋은, 손에 익은 에디터를 잘 활용하도록 하는 것이 중요하다.
- (9) 컴파일하여 결과 미리보기를 해보자.

### 3.2 라텍 문서의 구성 요소

TeX 문서는 텍스트와 텍스트를 보조하는 요소들, 그리고 수식 등으로 이루어진다. 텍스트를 보조하는 요소들에는 참조장치, 요약문, 각주, 그림과 표 등이 있다.

### 3.3 좀 긴 문서

```

\documentclass[a4paper,11pt]{oblivoir}

\usepackage{graphicx,xcolor}

\usepackage{amsmath,amssymb}

\begin{document}

\title{간단한 문서 샘플}
\author{홍 길 동}
\date{\today}
\maketitle

\begin{abstract}
이 문서는 가장 간단한 문서 샘플이다.
\end{abstract}

\tableofcontents

\section{서론}\label{sec:intro}
간단한 문서 샘플을 작성해보자.

\section{본론}

\subsection{인용문의 처리}
라텍에서는 인용 처리 방식이 매우 다양하다.
다른 아무런 설정이 없어도 인용을 잘 처리해 준다.\cite{HT}

또한 \pageref{sec:intro} 페이지의 \ref{sec:intro}\를
보면, 상호참조가 어떻게 구현되는가를 잘 알 수 있다.

\subsection{문서에 들어가는 것들}

수식은 다음과 같이 깔끔하게 처리된다.

\begin{equation}\label{eq:1}
P(z) = \prod_{n=1}^{\infty} E_p \left( \frac{z^{a_n}}{\right)
\end{equation}

```

그림은 외부 그림을 이용할 수도 있겠지만, 재미삼아 간단한 picture 그림~\ref{fig:1}\을 하나 넣어보겠다.

```
\begin{figure}
\centering
\setlength\unitlength{1pt}
\begin{picture}(88,60)
\multiput(55,40)(3,-3){10}{\circle{20}}
\multiput(85,10)(3,3){11}{\circle{20}}
\end{picture}
\caption{picture 환경의 그림}\label{fig:1}
\end{figure}
```

뭐, 이왕이면 표도 하나. 표~\ref{tab:1}\을 보라.

```
\begin{table}
\centering
\caption{표 샘플}\label{tab:1}
\begin{tabular}{|c|c|}
\hline
Hello & Good-bye \\ \hline
안녕 & 안녕 \\ \hline
\end{tabular}
\end{table}
```

식~\eqref{eq:1} 및 그림과 표는 문서를 더욱 풍부하게 해준다.\footnote{%  
여기 인용한 그림과 표는 \bnm{한글과 텍}에서 가져온 것이다.}

\section{결론}

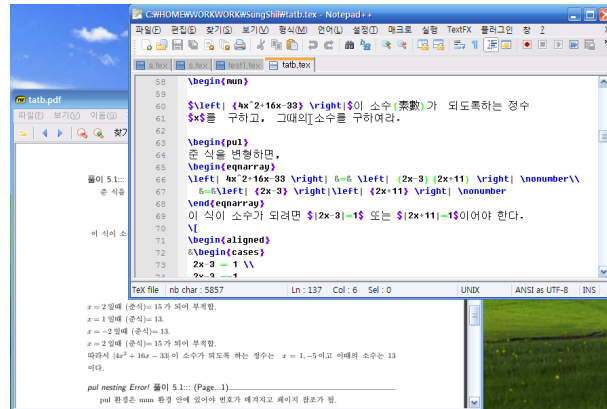
나쁘지 않은 것 같다.

```
\begin{thebibliography}{0}
\bibitem{HT} 김강수. ``한글 \LaTeX 의 현재와 미래,``  
온라인 문서.

\end{thebibliography}

\end{document}
```





## 4 중급자료의 첫걸음

### 4.1 라텍 사용하기의 두 모드

라텍 사용에는 두 가지 방식이 있다.

- ① 주어진 양식에 충실하게. 나의 목적은 글을 쓰는 것 또는 Presentation을 만드는 것이지 디자인이 아니다.
- ② 이왕이면 예쁘게 문서를 꾸며야 한다. 레이아웃, 폰트, 타이포그래피, 디자인 등을 나만의 것으로 하고자 한다.

일반적으로 ①의 경우는 특정 학술지에 기고하거나 졸업 논문을 쓰거나 하는 등 제한된 목적으로 쓰이고, 익혀야 할 것이 많지 않다. 수식을 자유롭게 쓸 수 있는 정도이면 됨.

그러나 ②의 경우는 디자인과 그 구현(TeX을 통한)의 문제가 된다. 이 때는 익혀야 할 것도 많고 알아야 할 것도 많다.

### 4.2 패키지의 활용

TeX의 가장 중요한 장점이, 상상가능한 거의 모든 문제에 대해 이미 답이 나와 있다는 것이다. 패키지를 활용하자.

예를 들면 페이지 레이아웃은 geometry 패키지를 이용한다. 페이지 스타일은 fancyhdr 패키지를 이용한다. 둘 이상의 그림을 나란히 배치하는 것은 subfloat 패키지를 이용한다.

거의 대부분의 자신이 원하는 기능, 모양이 어딘가 적당한 패키지로 존재할 것이다. 어떤 패키지가 어떤 기능을 하는지 모르기 때문에 문제가 되는데, 이런 것은 “물어서” 해결한다.

### 4.3 자신만의 레이아웃을 만들어보자

가장 쉽게 시작할 수 있는 곳은 레이아웃을 변경하는 것이다. 잘 만들어진 스타일을 분석하거나, 그 코드를 가져다가 자신이 원하는 방식으로 조금 변경하는 것이다.

chapter heading의 모양, 페이지 스타일(면주)의 모양 등을 원하는 대로 바꾸는 것은 이것이 시작이다. 먼저 preamble에 원하는 모양을 구현하는 명령을 만들어서 테스트한 후, 그것을 항상 사용할 수 있도록 .sty라는 확장명을 붙여서 저장해두면 그것이 소위 스타일 패키지가 될 수 있다.

다음 코드는 report 클래스의 장(chapter) 헤딩 모양을 바꾸기 위해 kosections-utf.tex 파일의 일부를 가져다가 변경한 것이다.

```

\makeatletter
\def\@makechapterhead#1{%
% \vspace*{50\p@}%
\vspace*{0pt}%
{\parindent \z@ \raggedright \normalfont
\ifnum \c@secnumdepth >\m@ne
\centering %%<-----
% \huge\bfseries
\LARGE\bfseries
\pre@chaptercnt\thechapter\post@chaptercnt
% \par\nobreak
% \vskip 20\p@
\fi
% \interlinepenalty\@M
% \Huge \bfseries #1\par\nobreak
\quad #1%
\vskip 40\p@
}}
\makeatother

```

#### 4.4 L<sup>A</sup>T<sub>E</sub>X 프로그래밍을 해보자

예전에 필자는 문제와 답을 함께 입력해두면 답만 모아서 따로 출력하는 기능을 구현해본 적이 있다. 대략 다음과 같은 모양.

```

%%%%%%%%%% b %%%%%%%%%%%
\newcommand\outputfilename{collectedsol.tex}
\newoutputstream{outputstream}\newinputstream{inputstream}
\IfStreamOpen{outputstream}{}{%
\openoutputfile{\outputfilename}{outputstream}%
}

\def\ERRMSG{}

\newcounter{munje}[section]\setcounter{munje}{0}
\newcommand\fmtmunje{\arabic{section}.\arabic{munje}}
\newif\ifInMunEnv\InMunEnvfalse

```

```

\newenvironment{mun}
  {\InMunEnvtrue\refstepcounter{munje}%
   \par\noindent{\bfseries [문제 \fmtmunje]\label{mun:\fmtmunje}}%
   \hfill\hyperlink{sol:\fmtmunje}{\heartsuit}\par}
  {\InMunEnvfalse\par\bigskip}

\newenvironment{pul}%
  {\ifInMunEnv\def\ERRMSG{}\else\def\ERRMSG{pul nesting Error! }\fi
   \let\thismunje\fmtmunje
   \addtostream{outputstream}{%
    \string\par\string\bigskip
    {\string\par\string\noindent\string\normalsize\string\sffamily
 \itshape\ERRMSG}풀이 \thismunje ::: (Page...\string\pageref
 {mun:\thismunje})\string\hypertarget{sol:\thismunje}{}}
   \string\hrulefill\string\par
   }
   \expandafter\writeverbatim\expandafter{outputstream}
  }%
  {%
   \endwriteverbatim
   \par
  }

\newcommand\printcollectedsol{%
  \clearpage
  \centerline{\LARGE 풀 이}
  \small
  \closeoutputstream{outputstream}
  \openinputfile{\outputfilename}{inputstream}
  \readstream{inputstream}
}
%% e

```

그다지 어렵지 않다. 중요한 것은 코딩 자체가 아니라 문제를 해결하는 ‘방법’ 또는 알고리즘이다. 이 방법을 실제 코드로 구현하는 것은 매뉴얼을 찬찬히 읽으면 다 나와 있다. (어디에 있는가, Know-Where 의 문제).

**4.5 폰트 문제**

최근 TeX 의 눈부신 발전은 — TeX 은 매우 오래된 프로그램이기도 하지만 최근 급격히 변모하고 있기도 하다 — 종래 TeX 의 한 특징처럼 여겨져 오던, 폰트 사용의 제약을 완전히 탈피했다. XeTeX 이라는 새로운 엔진의 등장으로 시스템 폰트를 자유롭게 사용할 수 있게 된 것이다.

이 문서는 XeTeX 을 이용하여 윤명조, 맑은 고딕 등의 트루타입 오픈타입 글꼴을 사용하여 작성하였다. 다음은 이 문서의 폰트 사용 설정 부분이다.

```

\setmainfont[Mapping=tex-text]{Minion Pro}
\setsansfont{Myriad Pro}
\setkormainfont(-윤명조340)(-윤명조330){-윤명조320}
\setkorsansfont{나눔고딕}
\setkormonofont{맑은 고딕}
\setmonoscale{.9}

```

#### 4.6 T<sub>E</sub>X 언어?

T<sub>E</sub>X이라는 언어는 그 자체로 문법을 갖추고 있어서 이 문법을 습득하는 것이 그다지 쉽지 않다. 이에 대한 한 가지 좋은 소식. LuaT<sub>E</sub>X이라는 새로운 엔진은 T<sub>E</sub>X 명령을 Lua 언어로 정의할 수 있게 하고 있다. 만약 Lua 스크립트 언어에 익숙하다면 T<sub>E</sub>X에 대해서는 최소한의 지식만을 가지고도 자유자재로 T<sub>E</sub>X commands 들을 정의하고 활용할 수 있게 할 수 있다.

다음 예는 간단한 견적서를 LuaT<sub>E</sub>X으로 출력하게 한 보기이다.(이호재 님이 작성하심.)

```

\begin{tabularx}{\textwidth}{|X|r|r|r|}
\hline
Description & Quantity & Unit Price & Amount \tabularnewline
\hline
%
\directlua0{
CurRow = 1
TotRow = 20
array = {
{"1st Driver Tests", 1, 2040},
{"1st Certification Tests", 1, 7650},
{"2nd Driver Tests", 1, 2040},
{"2nd Certification Tests", 1, 7650},
{"3rd Certification Tests", 1, 2040}
}
i = 1
total = 0
while array[i] do
array[i][4] = array[i][2] * array[i][3]
total = total + array[i][4]
str = array[i][1].."&"..array[i][2].."&"..
array[i][3].."string\\,\\string\\EUR &"..
array[i][4].."string\\,\\string\\EUR\\string\\tabularnewline\\string\\hline"
tex.print(str)
CurRow = CurRow + 1
i=i+1
end

```

```

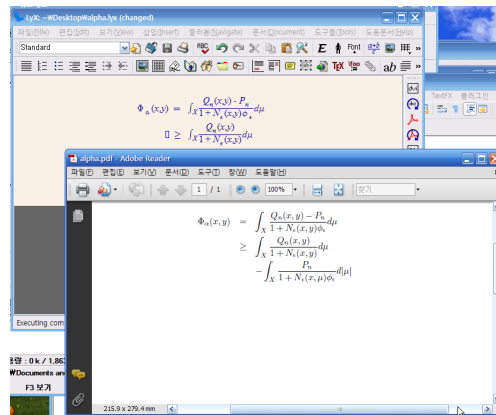
while CurRow <= TotRow do
str = "& & & \string\\tabularnewline\string\\hline"
tex.print(str)
CurRow = CurRow + 1
end
}
%
\multicolumn{3}{|r|}{TOTAL} &
\num{\directlua0{tex.print(total)}}\, \EUR\tabularnewline
\hline
\end{tabularx}

```

#### 4.7 또하나의 $\LaTeX$ 을 쓸 이유 --- Beamer

이번 강좌는  $\LaTeX$  중급자 강좌라서 제대로 다루지 못했지만, Beamer 강좌는 꼭 한 번 개최되기를 희망한다.

Beamer는  $\LaTeX$ 을 이용하여 PDF 발표자료 (Presentation)를 만드는 도구이다. MS Powerpoint가 하는 일과 비슷한 일을 한다.



## 5 Short Cuts

### 5.1 매뉴얼과 샘플

가끔 매뉴얼을 읽은 후에 문서 작성을 시작하겠다는 분들을 본다. 그러나, 이것은 효율적인 접근방법이 아니다.

- 먼저 좋은 샘플을 그대로 복사한 후 자신의 내용으로 채워넣는다.
- 아무리 좋은 샘플이라도 뭔가 바꾸고 싶은 것이 있게 마련.
- 바꾸고 싶은 것이나 그밖에 하고 싶은 것을 알아보기 위해서 매뉴얼을 이용한다. 즉, 매뉴얼을 레퍼런스처럼 이용하는 것.
- 어느 정도 되겠다고 생각하면 이제 연관된 내용을 찾으면서 매뉴얼을 훑어본다.
- 이 과정을 반복한다.

요컨대 좋은 샘플과 매뉴얼이 있으면 시작하기에 충분하다.

### 5.2 수식 입력 익히기

- $\TeX$ 의 수식 명령을 모두 아는 것은 불가능하다. 그러나 어느 정도는 알아두어야 한다.
- 수식 입력을 익히기 위해 반드시 읽어야 하는 문서. (1) amsldoc, (2) Mathmode
- LyX의 WysiWym 모드를 활용하여 원하는 수식을 만들어보면서 거기에 해당하는 명령이 무엇인지 알아본다.

## 6 L<sup>A</sup>T<sub>E</sub>X 문서 작성의 즐거움

얼마 전에 있었던 한 워크숍에서 Beamer에 대한 강연을 하신 분께서 매우 재미있는 말씀을 하셨다. 즉,

파워포인트는 시간이 갈수록 짜증스러워지는 데 반해, Beamer는 숙련되어갈수록 재미 있어진다.

고 하신 것. 또한,

내용이 부실해서 화려한 효과로 관객의 주의를 돌려야 하는 상황이라면 파워포인트를 이용하시고, 청중을 내용에 집중하게 하려면 Beamer를 이용하시라

일견 일리있는 말씀이다. 파워포인트도 매우 훌륭한 도구이므로 이런 식의 비교는 재미있게 말씀하시기 위해 한 것이겠지만, L<sup>A</sup>T<sub>E</sub>X을 기반으로 하는 대부분의 글쓰기 방식이 이와 유사하다.

즐거운 L<sup>A</sup>T<sub>E</sub>X 체험이 되시기를 빌며.