

An Introduction to L^AT_EX especially for GSHS students

Abstract

An-Introduction-to-LaTeX

논문작성을 위한 LaTeX 입문서를 LaTeX로 제작한 것입니다.

LaTeX Editor로는 TeXstudio를 추천합니다. 개인적으로 쓰고 있는 것이기도 해서...

<http://www.texstudio.org/>

TeXworks쓰다가 TeXstudio쓰면 정말 내가 왜 이걸 모르고 있었지 싶음...

TeX에서 한글을 사용하기 위해서는 koTeXlive를 별도로 설치해야 합니다.

www.ktug.org/xe/index.php?mid=install

용량이 1GB정도이니 주의 바람

웹사이트가 개설되었습니다! 새로운 버전이 생길 때마다 자동으로 업로드됩니다.

Download Page

찬양 1. HLETRD는 31기 선배들 중 L^AT_EX의 최고수입니다. 저희 32기에게 처음 L^AT_EX를 가르쳐 주시고 class 파일도 만들어주신 정민석 선생님께 감사드립니다.

우리의 기초 R&E는 수학 R&E가 아닌 LaTeX R&E이었습니다.

-익명

Contents

I	What is \LaTeX? How to install?	1
II	Basic functions of \LaTeX	1
1	header	1
2	High 가독성	1
3	<code>\ref</code>	1
III	making tables	1
1	easy table : using <code>\hline</code>	1
2	using <code>\multirow</code>	2
IV	Including images	4
1	first	4
2	using caption, label	5
V	Other math functions	5
1	Matrix	5
2	Pile	6
VI	Troubleshooter	6
1	Compile Error	6
1.1	Grammar error	6
1.2	Can't write on file	6
2	Broken text	6
2.1	<code>Ref[?]</code>	6
2.2	글자가 없는 경우	6
3	Label	6
VII	Other useful tips	7
1	Using <code>\newcommand</code>	7
2	TeXstudio - 'Convert to LaTeX'	7
VIII	Appendix	7
1	Fonts	7
2	Absolute Font Size in Standard Classes	7

List of Tables

1 sample table 4

List of Figures

1 *Sample figure* 5

I. What is L^AT_EX? How to install?

우리는 보통 논문이나 R&E보고서를 작성할 때 아래아한글이나 MS워드를 사용하곤 한다. 하지만 논문의 가독성 향상, 그림/표 넘버링과 간편한 리스트 작성을 위해 보통 대학원 이상에서는 L^AT_EX를 사용하곤 한다.

더욱 자세한 내용을 알고 싶다면 namu.wiki/w/LaTeX 을 읽어보길 추천한다.

TeX을 사용하기 위해서는 Editor가 필요하다. Windows의 경우 (무료) TeXworks, TeXstudio, TeXnicCenter 등이 있다. 개인적으로는 처음 배울때 TeXworks를 사용했었으나 뒤늦게 TeXstudio의 편리함을 깨우쳤다. TeXworks는 비추.

또한, 논문에 한글을 조금이라도 넣기 위해서는 kotexlive가 필요하다. www.ktug.org/xe/install 에서 다운받을 수 있다. 다만 용량이 1 GB 정도이니 데이터 주의.

참고로 kotexlive를 최초로 만드신 분이 경기과학고 출신이시다. ㅎㅎ

II. Basic functions of L^AT_EX

1. header

L^AT_EX문서도 C언어 프로그래밍과 같이 헤더를 필요로 한다. 사용할 라이브러리를 import해줘야 되고, 필요한 command들을 지정할 때 주로 문서의 맨 앞부분에 쓰고는 한다. 어디까지가 헤더라고 명확히 말할 수는 없지만, 이 문서의 헤더를 쓰면 다음과 같다.

2. High 가독성

L^AT_EX를 이용하여 문서를 작성하게 될 경우 작성 중 그리고 읽을 때 모두 가독성이 높다. 작성 중에는 엔터를 두번 쳐야 실제로 줄이 넘어간다. 강제로 줄을 바꾸고 싶을 경우에는 `\\`를 사용하면 된다.

3. `\ref`

이것이 I 절에서 말하였던 기능이다.

III. making tables

1. easy table : using `\hline`

```
\begin{table}[h]
  \begin{center}
    \begin{tabular}{|c|c|c|}
      \hline
    \end{tabular}
  \end{center}
\end{table}
```

이것은 표 삽입의 간단한 예제입니다.
표는 다음과 같이 삽입할 수 있다.

```

\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
학번&이름&특징\\
\hline
\hline
14012&홍길동&호부호형X\\
\hline
12320&전우치&도술에 재능이 있음\\
\hline
\end{tabular}
\end{center}
\end{table}

```

학번	이름	특징
14012	홍길동	호부호형X
12320	전우치	도술에 재능이 있음

```

=====
\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
학번&이름&특징\\
\hline
\hline
14012&홍길동&ㄱ ㄴ ㄷ ㄹ\\
12320&전우치&도술\\
\hline
\end{tabular}
\end{center}
\end{table}

```

학번	이름	특징
14012	홍길동	ㄱ ㄴ ㄷ ㄹ
12320	전우치	도술

2. using \multirow

```

\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline

```

```

\diagbox{학년}{반}&1&2\\
\hline
\hline
14012&홍길동&ㄱㅇㄹ\\
12320&전우치&도술\\
\hline
\end{tabular}
\end{center}
\end{table}

```

	반	1	2
학년			
14012		홍길동	ㄱㅇㄹ
12320		전우치	도술

```

=====
\begin{table}[h]
\begin{center}
\begin{tabular}{c|c|c}
\hline
\multicolumn{3}{c}{반장이름}\\
\diagbox{학년}{반}&1&2\\
\hline
\hline
\hline
14012&홍길동&ㄱㅇㄹ\\
12320&전우치&도술\\
\hline
\end{tabular}
\end{center}
\end{table}

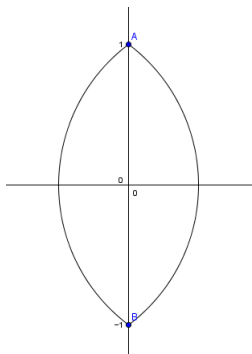
```

	반장이름		
	반	1	2
학년			
14012		홍길동	ㄱㅇㄹ
12320		전우치	도술

```

=====
\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c}
\hline
&\multicolumn{2}{c}{반장이름}\\
\hline

```



```

\diagbox{학년}{반}&1&2\\
\hline
\hline
14012&홍길동&ㄱㅇㅇ\\
\hline
12320&전우치&도술\\
\hline
\end{tabular}
\caption{1,2학년 반장}\label{tab1}
\end{center}
\end{table}

```

	반장이름	
반 \n	1	2
학년	14012	홍길동 ㄱㅇㅇ
	12320	전우치 도술

Table 1: sample table

IV. Including images

1. first

```

\begin{figure}
\begin{center}
\includegraphics[scale=0.4]{1.png}
\end{center}
\end{figure}
=====

```




=====

2. Pile

\$\$

```
f(x) = \left\{
\begin{array}{ll}
1, & x=3 \\
2, & x \neq 3
\end{array}
\right.
\right.
```

$$f(x) = \begin{cases} 1, & x = 3 \\ 2, & x \neq 3 \end{cases}$$

=====

VI. Troubleshooter

1. Compile Error

tex 문서가 컴파일되지 않는 경우에는 다음이 가능하다.

1.1. Grammar error

grammar error를 고쳐준다. 특히, 수학 전용 수식 명령어가 포함되어 있는데 수식 시작부분과 끝부분에 \$를 넣지 않았다면 넣어주도록 한다.

1.2. Can't write on file

Windows의 경우 OS 특성상 파일이 한 곳에서 이미 열려있을 경우 다른 곳에서 수정이 불가능하다. pdf파일을 닫고 다시 컴파일을 시도하도록 한다.

2. Broken text

2.1. Ref[?]

reference 기능을 사용할 때 컴파일 후 pdf 파일을 확인하면 대괄호 안에 ?가 되어있을수 있다. 이는 수정된 문서를 최초로 컴파일했을때의 문제로, 컴파일을 한번 더 해주면 해결된다.

2.2. 글자가 없는 경우

~ 와 같은 문자는 아무 생각 없이 그대로 넣어버리곤 한다. 그러면 글자가 나오지 않는다. \$\sim\$ 와 같이 넣어주도록 하자.

3. Label

```
\begin{figure}[h]
\begin{center}
\includegraphics[scale=0.5]{Taxicab Hyperbola.png}
```

```
\caption{2차원 택시 쌍곡선}\label{fig2}
\end{center}
\end{figure}
```

와 같이 사진 이름에 공백이 있으면 사진 이름이 같이 나오는 버그 발생.

VII. Other useful tips

1. Using `\newcommand`

지난 2014 기초 R&E논문작성 때 Taxicab Hyperbolloid를 여러번 쓰느라고 고생했었다. 하지만 `\ding`을 써주기만 하면 다음과 같이 나온다. 문서의 앞부분에 `\newcommand{\ding}{\textit{Taxicab Hyperboloid}}`와 같이 써주어야 한다. *Sample figure*

2. TeXstudio - ‘Convert to LaTeX’

TeX에 >와 같은 문자를 쓸 때 그냥 쓰면 깨지거나 에러가 난다. 따라서 일일이 `\textgreater`와 같이 변환해 주어야 된다. (심지어 방금 본 `\do` `\textbackslash`와 같이 써야 된다! 부들부들)

특히 정보 R&E의 경우 코드를 논문에 넣어야 하는 경우가 많은데, 각종 특수문자들을 일일이 LaTeX의 양식에 맞게 고치는 것이 매우 귀찮다!

찬양 3. Dev-C++에서는 ‘Export’-’to TeX’ 기능이 있다. 기능을 실행하면 cpp파일과 동일한 폴더에 tex파일이 하나 생겨 있을 것이며, 그를 compile하면 Dev-C++ IDE에서 보았던 그대로 코드가 나온다.

하지만 찬양 3와 같은 방법을 추천하지는 않는다. 코드 중 한 줄이 길어지게 될 경우 논문에서 잘리며(...), 필요 없이 코드에 색다른 폰트와 색을 넣어 놓는다. (코드의 가독성을 위해 색이 있어야 하는 것이 아니냐고 할 수도 있지만, 코드가 쉬워 보이게 된다는 단점이 있...다고 하는 분들도 계시다. ㄹㅇㅇㄹ) 그러기 위해 수없이 많은 package들을 include해야 되게 되므로 compile시간이 길어진다. 찬양 3와 같은 방법을 사용하게 될 경우 코드는 가장 마지막에 넣도록 하자.

koTeX의 경우 `\usepackage[hangul]{kotex}`과 Dev-C++에서 자동변환할 때 사용하는 package중 하나인 `\usepackage[ansinew]{inputenc}`와 상충한다. 이는 ansinew가 kotex와 충돌해서 그런데, 정확한 메커니즘은 모르겠으나 대충 유니코드 충돌 문제가 아닌가 추측된다. 해결 방법은 해당 package를 포함시키지 않거나 ansinew를 utf8로 바꾸면 정상적으로 컴파일된다. 다만 이에 대한 부작용은 아직 켜어보지 않아서 모른다. 유의하자.

또, 개인적인 의견이지만 TeXstudio에서 `ttfamily`(코드를 처리하는 float environment 부분)를 접어놓아도 그 앞에서 \$를 추가할 때 자꾸 `ttfamily` 안의 \$와 반응하여 접어놓은게 풀린다. 이것 어떻게 직접적으로 막을 수 있는지는 모르겠지만 Windows 운영체제에서의 TeXstudio 기준으로 Ctrl+Shift+M은 \$를 추가해주므로 해당 문제를 우회할 수 있다. 꼭 이쪽이 아니더라도 유용한 단축키이다.

찬양 4. TeXstudio의 토하나 편한 기능이 일일이 `textbackslash`할 필요 없이 Idefix-Convert to LaTeX 기능을 쓰면 이것처럼 바로 변환해줌.

...더이상 말이 필요한가요. texWorksEditor 말고 TeXstudio를 사용합시다 여러분

VIII. Appendix

1. Fonts

2. Absolute Font Size in Standard Classes

<code>\textrm{...}</code>	<code>textrmroman</code>	<code>\textsf{...}</code>	<code>textsfSans serif</code>
<code>\texttt{...}</code>	<code>texttttypewriter</code>		
<code>\textmd{...}</code>	<code>textmdmedium</code>	<code>\textbd{...}</code>	<code>textbdbold face</code>
<code>\textup{...}</code>	<code>textupupright</code>	<code>\textit{...}</code>	<code>textititalic</code>
<code>\textsl{...}</code>	<code>textslslanted</code>	<code>\textsc{...}</code>	<code>textscSMALL CAPS</code>
<code>\emph{...}</code>	<code>emphemphasized</code>	<code>\textnormal{...}</code>	<code>textnormaldocument font</code>

size	10pt{default}	11pt option	12pt option
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt